SZÁMÍTÁSTECHNIKAI

ALAP-

LEXIKON

I.

JODÁL ENDRE: ÁLTALÁNOS FOGALMAK

ALAPLAP KÖNYVEK 5

Számítástechnikai alaplexikon I.

Jodál Endre
Általános fogalmak

Számítástechnikai alaplexikon I.

Jodál Endre

Általános fogalmak

CÉDRUS KIADÓ

Sorozatszerkesztő: Faklen Pál
Szaklektorok: Barna László
Farkas Ernő
Nyelvileg ellenőrizte: dr. J. Gallasy Magdolna

A szótár munkálataiban közreműködtek:
Szakonyi Gergely
Zoltai Péter

ISSN 0866–434X
ISBN 963 7429 018 összkiadás
ISBN 963 7429 042 I. kötet

© Jodál Endre, 1991

Szerkesztő: Tamás Zsuzsa
Felelős kiadó: Sebestyén Ilona
Felelős vezető: Vékony Tamás
Cédrus Informatikai Részvénytársaság

Tördelés, nyomdai előkészítés: **PRINTSELF** Kft.
Felelős vezető: Dr. Kassay Árpád

Nyomtatás: Sprint^R – Hungaroprint
Felelős vezető: Tóth Éva
Készült a Stoffel & Tsai. gondozásában

Tisztelt Olvasó!

Az Alaplap Könyvek sorozaton belül egy új sorozatot indítunk útjára, az ALAPlexikon sorozatot. Annak ellenére, hogy az elmúlt években több számítástechnikai szótár látott napvilágot, úgy gondoljuk, hogy a mi sorozatunk hiánypótló. Tapasztalataink szerint ugyanis a számítástechnika alkalmazásának számos részterülete – így például a PC-k rohamosan bővülő világa – mostohán van ellátva lexikon jellegű kézikönyvekkel. Az elmúlt évtizedben a számítástechnika rohamosan terjedt, nagyon sok vállalatnál már számítógépen dolgoznak, egyesek csak szövegszerkesztővel, mások táblázatkezelőkkel is, sokan autodidakta módon tanulják a gép használatát – nekik különösen hasznos lehet lexikonsorozatunk.

A *Számítástechnikai alaplexikon* abban különbözik az eddig megjelent többi szótártól, hogy értelmező, tehát a szerző minden kifejezést megmagyaráz. A Kiadó törekvése elsősorban az volt, hogy olcsó áron minél több érdeklődőhöz eljuttathassa a könyvet. Ennek érdekében a lexikon szóanyagát témakörökre bontottuk, és most e témakörök közül az elsőt, az Általános fogalmak kötetet tartja kezében az Olvasó. A lexikon további kötetei 1992-93-94-ben folyamatosan jelennek meg. Aki e lexikont kézbe veszi, nyíllal jelzett utalásokat talál benne, ezek az utalások olyan módon vannak a szócikkekben elhelyezve, hogy ha egy kifejezést, szót nem értünk, az utalások nyomán végiglapozva a könyvön az egész vele kapcsolatos témakört megismerhetjük. Ennélfogva a lexikon tanulásra is alkalmas.

Első kötetünk szerzője, Jodál Endre már 1990-ben Bukarestben a Kriterion Kiadónál megjelentette *Számítástechnikai kislexikon*át, melyet annak idején a szakma itthon is melegen üdvözölt és nagy érdeklődéssel forgatott. Jodál Endre e kötet alapján, ezt kibővítve és természetesen modernizálva, átdolgozva készítette el a Cédrus Kiadó számára az ALAPlexikon első kötetét.

A szerző arra törekedett, hogy magyar megfelelőt adjon azoknak az angol szavaknak, amelyek még nem honosodtak meg nyelvünkben, ezáltal is mintegy befolyásolni akarván a közvéleményt, hogy ha lehet, a számítástechnika nyelvén is magyarul beszéljünk. Éppen ezért hiába keresi az Olvasó a *fájl* szót a kötetben, ezen az Előszón kívül máshol nem bukkanhat rá (ebben a formában, fonetikusan írva). Ez a szerző határozott kívánsága volt, miután a *file* fogalmát tökéletesen kifejezi az *állomány* magyar szó, és nem indokolt a fonetikusan ejtett angol szónak sem teret engedni. Az állomány szónál maradva, azért tettük előtte szögletes zárójelbe az adat szót, mert régebben a *file*-t adatállománynak nevezték. Nagyon sok szakmai könyvben és leírásban még így szerepel, bár a rövidebb forma már kiszorította, de gondolnunk kell az „újszülöttekre", az [adat]állomány formában könnyebben megtalálják és azonosítani tudják.

A kezdőkre, illetve tanulni vágyókra gondoltunk a kötet végén lévő angol-magyar visszakeresővel is, főleg azokra, akik a fogalmaknak nem a magyar, hanem az angol elnevezésével találkoznak. Ezzel az angol visszakeresővel így is eljutnak a kötet minden magyar szavához. Az angol-magyar szójegyzékbe vettük fel azokat az angol rö-

vidítéseket is, amelyeknek kifejtése, magyarázata a fogalom magyar megfelelőjénél található, például *DP* (Data Processing) helyett *adatfeldolgozás* a címszó (aki a DP-t keresi a szójegyzékben, onnan megtudja, hogy a lexikonban az *adatfeldogozás* címszónál találja a magyarázatát). De önálló címszóként találhatók a lefordíthatatlan, illetve a magyar nyelvben is így használt angol betűszavak, például a *COBOL*, az *EAN* stb.

A szerzővel együtt arra törekedtünk, hogy a címül szolgáló általános fogalmak mind legyenek benne a kötetben. Sok nagygépes kifejezés és eljárás megtalálható a könyvben. Ezeket ma már ritkábban használják, mégis benne kellett hagynunk a történetiség kedvéért, és azért, mert régebbi szakkönyvekben esetleg ilyen utalások is találhatók, és igy ezeket is meg kellett magyaráznunk.

Talán kevesli majd az olvasó a PC világára vonatkozó kifejezések számát. Ezzel kapcsolatban a Kiadó tervezi, hogy egy, a PC-ről szóló kötetet is meg fog jelentetni.

A szavakat a Magyar Tudományos Akadémia Helyesírási Szabályzata alapján írtuk, tehát figyelembe vettük a jelenleg érvényes szabályokat. Szeretnénk, hogy ha ezzel a lexikonnal nemcsak az ismeretanyagot, hanem a *helyesírást* is sugallhatnánk, mintegy lépéseket téve afelé, hogy a magyar számítástechnikai folyóiratok, szakkönyvek nyelvezete egységes legyen. Ez természetesen nem jelenti azt, hogy az általunk használt szóalak vagy írásmód a legjobb. De miután több szaklektor, számítástechnikai szakember szóhasználata, helyesírási felfogása tükröződik a kötetben, szeretnénk ezt kiindulási pontul felajánlani a szakmának.

Ugyanezen okból kifolyólag a tervezett utolsó kötetben szeretnénk a kiejtést is „szabályozni", illetve arra tennénk kísérletet, hogy a különböző angol szavakat, amelyek már meghonosodtak a magyar nyelvben, a szakma szintén egységesen ejtse ki. (Például van, aki az *UNIX*-ot *junix*-nak, s van, aki *unix*-nak ejti, s a *DOS* egyeseknél *dos*, másoknál *dosz*, sőt *doz*.) Az eltérő kiejtés alkalmanként helyesírási különbözőségeket is okoz.

Észrevételeiket, megjegyzéseiket, kiegészítő javaslataikat a Kiadó szívesen fogadja, mert azok a következő kötetek szerkesztésekor hasznos segítséget jelenthetnek.

A Kiadó

A, Á

A – a →hexadecimális ábrázolás jelkészletének eleme, amely a decimális 10 értéknek felel meg (bináris megfelelője: 1010).

ábécé (alphabet) – valamely természetes v. mesterséges nyelv mondatait felépítő szimbólumok rendezett halmaza.

ablak (window) – 1. →Párbeszédes üzemmódban dolgozó rendszereknél használt módszer a felhasználóval tartott kapcsolat megvalósítására: a képernyő valamely téglalap alakú területe, ahol a végrehajtás alatt álló →program kivetíthet üzeneteket, →menüket, →segélyállományok lapjait stb. Egy időben több ~ is jelen lehet a képernyőn, sőt egymásra is lehet őket vetíteni. – 2. A →grafikus adatfeldolgozásban használt módszer, amely lehetővé teszi, hogy a logikai alakzatnak csak egy részét jelenítsék meg a képernyőn. Az ~ tkp. a megjelenítendő modelltérnek egy jól meghatározott – két- v. háromdimenziós – résztartománya. Gyakorlati megfontolásokból a legtöbb grafikus rendszernél az ~ négyszögletes tartományként adható meg, amelynek oldalai párhuzamosak a modelltér, ill. a rasztertér koordinátatengelyeivel. Az alakzatnak az ~ban látható képe rendszerint nem tölti ki a képernyő teljes felületét, így elegendő szabad hely marad további ~ok, ill. más képmezők részére, ahol a feldolgozással kapcsolatos egyéb információk jeleníthetők meg. – 3. →Adatátviteli hálózatok →protokolljában a fogadó által a küldőnek nyújtott terület, amelyben üzenetek, átviteli paraméterek és vezérlési információk helyezhetők el (pl. az átvitel sebességére, az átvitt adatmennyiségre stb. vonatkozóan).

abszolút betöltőprogram (absolute loader) – →bináris betöltőprogram

abszolút bináris forma (absolute binary form) – számítógépes programoknak olyan bináris kódolású formája, amelyben azok végrehajtás céljából az →operatív memóriában megtalálhatók, ezért közvetlenül végrehajtható formának is nevezik. Eltekintve a →bázisregiszterrel, ill. →indexregiszterrel képzett →címektől, a program minden hivatkozása →abszolút címre vonatkozik. Az ~ a →tárgyprogramokhoz tartozó tulajdonképpeni bináris szövegen kívül nem tartalmaz járulékos információkat.

abszolút cím (absolute address) – az →operatív memóriában v. valamely címezhető →adathordozón rögzített információegység tényleges, fizikai helyének általában számmal kódolt →címe, amelyet a →központi vezérlőegység közvetlenül értelmezhet. Az operatív memóriánál az ~ annak a rekesznek a sorszáma, amely a végrehajtandó utasítást, ill. a feldolgozandó adatot tartalmazza. A címezhető →mágneslemeznél a →fizikai bejegyzés ~ét (C,S) számpár jelölheti, ahol a C a →cilinder, az S a cilinderen belüli →sáv számát rögzíti.

abszolút hiba (absolute error) – az a →hiba, amelynek mértéke a számítandó mennyiség valódi és számított értéke közötti különbség.

abszolút kifejezés (absolute expression) – az →asszembler nyelven kódolt programokban valamely operandus →címének meghatározására használatos olyan aritmetikai kifejezés, amelynek értéke a program →áthelyezésénél nem módosul (szemben az →áthelyezhető kifejezéssel). Az ~ben az áthelyezhető elemek páros számban, csak összeadásban és kivonásban szerepelhetnek, s páronként ellenkező előjellel csoportosíthatóknak kell lenniük (két áthelyezhető elem különbsége ugyanis mindig abszolút értéket ad, így a kifejezés abszolút jellege biztosítva van).

abszolút kódolás (absolute coding) – valamely →algoritmusnak közvetlenül →gépi nyelven történő előállítása az →operatív memóriában. Az így készülő program minden további átalakítás nélkül végrehajtható.

absztrakt [adat]típus (abstract [data] type) – egyes korszerű →programozási nyelvekben (pl. →Ada) az a lehetőség, hogy a programozó a saját →[adat]típusait meghatározza. E meghatározásnak vonatkoznia kell nemcsak a felvehető értékekre, hanem az ezekkel elvégezhető műveletekre is. Gyakorlati megvalósítása általában transzparens a felhasználó számára. Az ~ jelentősége az, hogy a felhasználó adatainak ábrázolásától független programok írhatók, s a konkrét ábrázolásmód megváltoztatása nincs kihatással a programokra.

absztrakt program (abstract program) – a →strukturált programozással kapcsolatos fogalom. A készítendő programot megvalósító modellsorozat minden eleme egy teljes és

bizonyítottan helyes ~, amely a neki megfelelő →absztrakt számítógépen hajtható végre. Az adott szinten működő absztrakt számítógép elemi utasításkészletéhez a megfelelő ~ primitív utasításai tartoznak. Ezekre vonatkozóan – a következő szint megvalósításához – pontos szemantikai meghatározásokat kell adni (→lépésenkénti finomítás elve).

absztrakt számítógép (abstract computer) – a programozás technológiájában használatos fogalom. A valóságban nem feltétlenül létező, olyan rendszert jelent, amely (a programok tervezésének bizonyos szintjein) a legjobban megfelel az adott feladat leírásának és megoldásának. Alapvető jellemzője az a nyelv, amelyet az ~ →gépi nyelvének lehet tekinteni, természetesen az illető nyelv pontos szintaktikai és szemantikai meghatározása ismeretében (→strukturált programozás).

absztrakt szintaxis (abstract syntax) – az a (J. McCarthy által 1962-ben bevezetett) fogalom, a →programozási nyelvek →szintaxisának leírására, amelyben az értelmezés az elsőrendű cél, nem a programok tényleges megjelenési formája (→konkrét szintaxis).

Ada – az egyik legmodernebb magas szintű algoritmikus programozási nyelv, amelyet a J. D. Ichbiah vezette tervezőcsoport valósított meg 1974 és 1980 között a CII-Honeywell/Bull franciaországi laboratóriumában, majd 1983-ban standardizálták az ANSI-nál. Kiindulópontja a →Pascal, így az →erősen típusos nyelvek családjába tartozik, s magában foglalja az elméleti számítástechnikai kutatások legfontosabb eredményeit. Legfontosabb alkalmazási területe a →beágyazott rendszerek előállítása, de egyaránt használható tudományos-technikai, gazdasági-pénzügyi, rendszerprogramozási stb. területen is. Hatékony eszközöket nyújt a felhasználóknak a valós idejű konkurens folyamatok magas szintű programozására, a →kivételek biztonságos kezelésére, lehetőséget ad a →külön fordításra és az →információrejtésre (→csomag). Elsőrendű célkitűzése a teljes gépfüggetlenség elérése, a tényleges →hordozhatóság megvalósítása. Rendszerprogramozást támogató eszközei révén alkalmas teljes →programozási környezet kifejlesztésére.

adagolt feldolgozás (batch processing) – →kötegelt feldolgozás

adaptív csatornakiosztás (adaptive channel allocation) – az →adatátviteli hálózatokban használt eljárás, amely a mindenkori igényeknek megfelelően osztja ki az adott →csatorna átviteli kapacitását a különböző erőforrások között.

adásvége karakter (end-of-transmission character) – a kölcsönös szinkronizálással vezérelt →szinkron adatátviteli eljárásokban használt vezérlő jel. Az →üzenetkapcsolásos, ill. a →csomagkapcsolásos hálózatokban rendszerint ~ bontja az összekapcsolást. A →lekérdezéses szervezésben az ~ az egész átviteli rendszert alapállapotba hozza, vagyis az adatátviteli vezérlőegység adóüzemmódba és vezérlési állapotba kerül, és megkezdheti az adatállomások lekérdezését.

adat (data) – az →információnak a számítógépes rendszerben használt konkrét megjelenési formája, vagyis minden olyan jel v. jelkombináció, amely valamilyen →programozási nyelven írt utasítás operandusa. Az információfeldolgozás bizonyos fázisaiban nem veszik figyelembe az információk tényleges jelentését, hanem azok helyett az ~okkal végzik el a kívánt műveleteket. Az ~ot a számítógép szempontjából a hozzárendelt memóriarekesz(ek) →címe és hosszúsága, valamint a szerkezetére vonatkozó információk jellemzik. A magas szintű programozasi nyelvekben az ~ot v. a hozzárendelt – implicit v. explicit – →deklaráció v. – ha változatlan értékkel rendelkezik az egész feldolgozás alatt – az adott értéket jellemző jelkombináció képviseli (→logikai-fizikai leírás).

adatabsztrakció (data abstraction) – programozástechnikai szempontból az adatfeldolgozással és -kezeléssel kapcsolatos módszer. Lényege az, hogy a különböző szintű felhasználók szempontjából az adatok jellemzői közül csak egy v. csak néhány játszik szerepet, a többi rejtve maradhat. Az adatokkal kapcsolatban végzett absztrakció a különböző →[adat]típusok meghatározásához vezetett, ezek az adatszerkezetek elméletének alapvető fogalmai. Az ~ elemi fokon a legtöbb magas szintű programozási nyelvben szerepel. Pl. az →egész [adat]típus fogalom használatakor a programozónak nem kell törődnie sem az egész számok ábrázolási módjával, sem az azokra vonatkozó gépi uta-

sítások formájával. A megfelelő eljárásokat a →fordítóprogram, ill. a számítógép →architektúrája bocsátja a felhasználó rendelkezésére.

[adat]állomány (file) – →lineáris adatszerkezeteknek külső →adathordozón kialakított rendezett halmaza a nyilvántartás, a hozzáférés és a megőrzés szempontjából. Más megközelítésben: az egymással kapcsolatban álló objektumok (tárgyak, elemek) rendezett halmaza; ezek mindegyike rendelkezik legalább egy olyan tulajdonsággal, amelyhez értéket lehet hozzárendelni. Az ~t alkotó objektumok az →entitások, megjelenési formájuk pedig a →logikai rekordok. Az ~ tehát a logikai rekordok rendezett halmazaként is meghatározható, s ebben az értelemben a logikai ~t jelenti. Az adathordozón rögzített →fizikai bejegyzések rendezett halmaza viszont a fizikai ~t képezi (→logikai-fizikai leírás).

Az adatszervezés módszere szerint az ~ok négy alapvető változata ismeretes:

1) Szerkezet nélküli ~nál a fizikai bejegyzések úgy helyezkednek el egymás után az adathordozón, hogy sem a bejegyzések azonosítója és ~ban elfoglalt helye között, sem a fizikailag egymás után következő bejegyzések között nincsen semmiféle logikai kapcsolat. A bejegyzések folyamatosan követik egymást az adathordozón, s feldolgozásuk csak sorosan, kizárólag fizikai szinten végezhető.

2) Szekvenciális ~nál az egymást követő objektumok bizonyos tulajdonságai között egyértelmű logikai kapcsolat áll fenn. A logikai rekordok elhelyezése az ~ban e tulajdonság(ok) sorrendjében történik. Előállítása és feldolgozása mindig szekvenciális.

3) Indexszekvenciális ~nál az egymást követő objektumok tulajdonságai közötti kapcsolat az objektumok helyét is befolyásolja. Előállítása mindig szekvenciálisan történik, feldolgozása lehet szekvenciális és közvetlen is.

4) Direkt szervezésű ~nál az objektumok tulajdonságai egyértelműen meghatározzák azok helyét az ~ban. Az előállítás és a feldolgozás egyaránt közvetlen módú.

[adat]állomány-aktivitási hányados (file activity ratio) – valamely →[adat]állomány egyszeri aktualizálására jellemző érték, a mágneses adathordozók megválasztására és a fizikai adatszervezés típusának kiválasztására használják. Értékét a ténylegesen érintett, ill. az összes beolvasott rekordok számának hányadosa adja. Általánosan elfogadott vélemény szerint, ha az ~ meghalad egy bizonyos értéket (kb. 25%-ot), az adatszervezés típusa rendszerint szekvenciális, így előnyösen alkalmazható az adatok tárolására a viszonylag olcsó →mágnesszalag. Ha az ~ a fenti érték alatt van, célszerű a közvetlen elérésű adathordozó (→mágneslemez) használata, ami drágább ugyan, de elkerülhetővé teszi a hosszadalmas keresési műveleteket.

[adat]állomány-aktualizálás (file updating) – az →aktualizálási művelet végrehajtása →[adat]állományokon. Rendszerint nem érinti azok fizikai és logikai szerkezetét, csak a tárolt információk tartalmára vonatkozik. Az ~ alapműveletei: meglévő rekordok javítása, módosítása, törlése, ill. új rekordok beírása az [adat]állományba.

[adat]állomány-azonosítás (file identification) – az →[adat]állomány-kezelő rendszer által végrehajtott műveletsor: a feldolgozandó [adat]állományt az →adathordozón lokalizálja, állapotát és jellemzőit ellenőrzi és azt feldolgozásra alkalmas állapotba hozza. Az azonosításhoz a felhasználónak meg kell adnia egyrészt az adathordozóra vonatkozó információkat (típus, meghajtó →periferikus mű, kötetazonosító stb.), másrészt az [adat]állományt jellemző információkat (szervezési mód, →generációszám, →verziószám, hozzáférési mód, →blokkolási paraméterek, rekordformátum, pufferkezelés stb.). Az azonosítási információk leírására rendszerint az →asszembler nyelvben rendelkezésre álló →makroutasítások, a →munkavezérlő nyelv vezérkártyái szolgálnak (→kötegelt feldolgozásnál), ill. speciális parancsok használhatók (a →párbeszédes üzemmódban dolgozó rendszereknél). Ha az [adat]állományt elő is kell állítani, az ~ magában foglalja még az adathordozó →szabványos címkékkel történő ellátását is. Az ~nak sikertelensége a felhasználói program rendellenes befejezéséhez vezet.

[adat]állomány-karbantartás (file maintenance) – az →[adat]állományokon végrehajtott olyan műveletek összessége, amelyeket a biztonságos, gyors és megbízható feldolgozás érdekében végeznek el. Ilyen mű-

velet pl. az →indexszekvenciális v. →direkt szervezésű [adat]állományok időnkénti átszervezése, a logikailag törölt →rekordok fizikai törlése, másolatok készítése biztonsági okokból stb. Az ~ elvileg különbözik az →[adat]állomány-aktualizálástól, mert annak célja az [adat]állomány olyan irányú megváltoztatása, hogy az mindig az aktuális, naprakész információkat tartalmazza. Az ~ soha nem változtat az [adat]állomány logikai állapotán, csupán a fizikai adatszervezést megvalósító másodlagos adatokat érinti.

[adat]állomány-keresés (file searching) – valamely →[adat]állománynak az →újrahasználható adathordozón történő lokalizálása. A művelet az [adat]állomány jellemzőinek – azonosító, tulajdonos, →generációszám, →verziószám stb. – segítségével valósul meg, amelyek a →szabványos címkékben találhatók.

[adat]állomány-kezdőcímke (file header label, HDR) – az →[adat]állomány egyértelmű azonosítására szolgáló információk összessége. Szabványos formája 80 →bájt hosszúságú blokk, amelynek első négy karaktere szolgál a címke azonosítására (HDR+sorszám). Tartalma függ az →adathordozó típusától (→mágneslemez v. →mágnesszalag), de bizonyos információk függetlenek tőle. A HDR1 tartalmazza az [adat]állomány nevét, →generációszámát, →verziószámát, az előállítás dátumát, a megőrzési dátumot, jelzi az adatok titkosságát stb. A HDR2 tartalmazza a →fizikai bejegyzések típusát (fix, változó v. definiálatlan forma), a bejegyzések hosszát, a →blokk hosszát stb. A →szabványos címkék feldolgozása az →operációs rendszer feladata. A felhasználói ~ (azonosítója UHL) lehetőséget ad a felhasználónak arra, hogy további információkat is tároljon, de feldolgozásukról magának kell gondoskodnia, ez azonban az operációs rendszertől is függ.

[adat]állomány-kezelés (file management) – a →logikai [adat]állományokra v. a bennük szereplő →rekordokra vonatkozó műveletek összefoglaló megnevezése. Három fő csoportja:

1) A szerkesztés jellegű műveletek; az [adat]állományra mint logikai egységre vonatkoznak ([adat]állomány létrehozása, másolása, megszüntetése, újraszervezése stb.).

2) A karbantartás jellegű műveletek; az [adat]állomány fizikai szerkezetére vonatkoznak, az adatszervezést megvalósító másodlagos információk megváltoztatását teszik lehetővé (logikailag törölt rekordok tényleges törlése, →indexszekvenciális v. →direkt szervezésű [adat]állományok átszervezése stb.).

3) A feldolgozás jellegű műveletek; az [adat]állományban rögzített rekordokra mint logikai együttesekre vonatkoznak (adott feltételnek megfelelő rekord kikeresése, aktualizálása és visszahelyezése az eredetileg elfoglalt logikai helyének megfelelően).

[adat]állomány-kezelő rendszer (file management system) – valamely →operációs rendszer adatkezelő →szubrutinjainak összessége, amelyek a →logikai [adat]állományok kezelésével és feldolgozásával kapcsolatos műveleteket végzik. Mivel a számítógépek felhasználásában az adatfeldolgozási feladatok igen fontos szerepet játszanak, az operációs rendszerek értékének, színvonalának egyik lényeges mutatója az ~ szolgáltatásainak minősége. A rendszer legfontosabb funkciói: [adat]állományok létrehozása, ill. megszüntetése, megnyitása, ill. lezárása, →szabványos címkék kitöltése, ill. azok tartalmának ellenőrzése, →logikai rekordok →blokkokba tömörítése, ill. a blokkok tördelése, a rekordok keresése, írása, ill. olvasása stb.

[adat]állomány-kiosztási táblázat (file allocation table) – IBM-szabvány szerint szervezett →mágneslemezek helykihasználási információit tartalmazó táblázat. Legtöbbször 2 példányban is tárolják (arra az esetre, ha az egyik megsérülne v. olvashatatlanná válna). →Hajlékonylemezeken az ~ rendszerint a 0. oldal 0. →sávjának 2. →szektorán kezdődik, rögtön a →betöltőszektor után. Az →operációs rendszer az ~on keresztül tartja számon az adatterület felhasználását, ugyanis ennek minden egyes bejegyzése egy sajátos kód segítségével jelzi, hogy melyik szektort használják, melyik szabad v. melyik használhatatlan lemezhiba miatt.

[adat]állomány-kivonatolás (file inventory) – valamely →[adat]állományban található →logikai rekordok kinyomtatása olyan formában, amelyet a felhasználó közvetlenül értelmezhet. A legtöbb →operációs rendszer rendelkezik olyan segédprogrammal, amely

az ~t →nyomtató segítségével elvégzi (bináris, hexadecimális v. alfanumerikus formában). Igen fontos segédeszköze a programtermékek belövésének, mivel lehetőséget ad az [adat]állományokon végrehajtott módosítások ellenőrzésére.

[adat]állomány-létrehozás (file creating) – →[adat]állományokra vonatkozó, szerkesztés jellegű művelet. Legfontosabb funkciói: a megfelelő →adathordozó kiválasztása és lefoglalása, a szervezési információk feldolgozása és rögzítése a →szabványos címkékben, a →logikai rekordok elhelyezése az [adat]állományban a fizikai szervezésnek megfelelően.

[adat]állomány-lezárás (file closing) – a →logikai [adat]állományok feldolgozás utáni kezelése. Fontosabb műveletei: az →[adat]állomány-zárócímkék felírása (ill. ellenőrzése), az esetleges későbbi feldolgozásokoz szükséges információk rögzítése, a feldolgozást követő esetleges igények kiszolgálása (pl. a →periferikus mű felszabadítása v. foglaltsági állapotának fenntartása, a →mágnesszalag visszacsévélése stb.). A kimeneti [adat]állomány csak lezárásának sikeres elvégzése nyomán állhat rendelkezésre későbbi feldolgozások számára.

[adat]állomány-megnyitás (file opening) – →logikai [adat]állományok feldolgozásához szükséges kezdeti tevékenységek összefoglaló megnevezése. Idetartoznak a következő műveletek: az →adathordozó, ill. az →[adat]állomány azonosítása, a →szabványos címkék írása, ill. ellenőrzése, az [adat]állomány szervezésére vonatkozó információk rögzítése (→logikai rekordok típusa és hossza, →fizikai bejegyzések hossza, →blokkolási tényező, →pufferek kezelése stb.), az esetleges adatvédelmi információk ellenőrzése stb. Az ~ az →operációs rendszer adatkezelő szubrutinjainak a feladata, a felhasználó speciális utasítások segítségével közölheti az operációs rendszerrel az ~ra vonatkozó információit. Az [adat]állomány csak sikeres megnyitás után állhat a felhasználó rendelkezésére.

[adat]állománynév (file name) – valamely →[adat]állomány azonosítására szolgáló alfanumerikus karakterlánc. Képzésére a konkrét →[adat]állomány-kezelő rendszer ad pontos szabályokat.

[adat]állománynév-kiterjesztés (file name extension) – egyes →operációs rendszerekben (pl. →UNIX, →MS-DOS stb.) az →[adat]állományok azonosítására használt minősítő, amely az →[adat]állománynévvel együtt képezi az egységes azonosítót. Rendszerint 1-3 alfanumerikus karakterből áll, az [adat]állománynévtől egy pont választja el, s két alapvető funkciót lát el: egyrészt megkülönbözteti egymástól a hasonló nevű [adat]állományokat, másrészt emlékeztető kódként utal(hat) az [adat]állomány jellegére és/vagy tartalmára.

[adat]állomány-zárócímke (end-of-file label) – az →[adat]állomány végét jelző →szabványos címke, amelynek tartalma megegyezik a megfelelő →[adat]állomány-kezdőcímke tartalmával. Szabványos formája 80 →bájt hosszúságú blokk, amelynek első négy karaktere szolgál a címke azonosítására (EOF+sorszám). →Visszafelé olvasáskor értelemszerűen az itt tárolt információk alapján történik az [adat]állomány azonosítása. A szabványos címkék feldolgozása az →operációs rendszer feladata. A felhasználói ~ (UTL) lehetőséget ad a felhasználónak arra, hogy további információkat is tároljon, de feldolgozásukról magának kell gondoskodnia.

adatállomás (terminal station) – az →adatátviteli hálózatok vonatkozásában minden olyan készülék, berendezés, amely adatok küldése v. vétele, esetleg mindkettő céljából kapcsolódik a hálózathoz. Az ~ lehet számítógépes (amikor az adatátviteli funkciók mellett helyi feldolgozásokat is biztosító kisszámítógép működik), és lehet adatvégállomás (tipikus képviselője a →terminál), amely kimondottan csak adatcserére alkalmas.

adatáramlás-vezérlés (flow control) – tárolókat használó →adatátviteli hálózatokban alkalmazott vezérlési eljárás, amely az adatátvitel sebességét a fogadhatóság szintjére korlátozza, s így elháríthatja a →torlódást. Két alapvető formája van:

1) végponttól végpontig terjedő ~nél a célállomás fogadási kapacitásának rendelik alá az adatátviteli sebességet (a teljes hálózat különböző szakaszain ez a sebesség változhat);

2) lépésről lépésre haladó ~nél az egyedi csomópontok fogadási kapacitása a mérvadó.

adatátviteli hálózat (data communication network) – digitális információk továbbítására alkalmas távközlési hálózat, távolsági adatátvitelt biztosító eszköz: légvezeték, kábelek, rádióösszeköttetések, távközlési műholdak, valamint az ezekhez tartozó kapcsolóközpontok, átalakítók és egyéb átviteli berendezések összessége. Az →adatállomások közötti összeköttetések lehetnek állandóak és közvetlenek is, de legtöbbször kapcsolóközpontok hozzák létre több független adatvonalszakasz átmeneti egymáshoz rendelésével.

adatátviteli művelet (data transfer operation) – minden olyan művelet, amelynek révén valamely adatrögzítésre alkalmas eszközről adatok kerülnek át más eszközre. A felhasználói programozó szempontjából csak a kiindulási és az érkezési →címek fontosak, az adatút tényleges kijelölése és az ~ vezérlése az →operációs rendszer feladata.

adatátviteli sebesség (data transmission rate) – az a sebesség, amellyel az adatátviteli vonal az adatokat továbbítani képes. A gyakorlatban (az adatvonalakra) igen gyakran az egy másodperc alatt átvihető elemi vonalállapotok számával adják meg az elérhető maximális ~et. Ha az átvihető legrövidebb vonalállapot „T" másodperc tartamú, akkor az ~et a következő képlet adja:

$$V = 1/T \text{ [baud]}$$

Egyetlen vonalállapot egy →bitnél több információt is hordozhat, ezért a bit/sec-ban mért ~ csak bináris átvitelnél egyezik meg a →baudban mért sebességgel:

$$V\text{[bit/sec]} = \log_2 n \cdot V \text{ [baud]}$$

ahol n a vonalállapotok lehetséges értékeinek száma, s így ennek kettes alapú logaritmusa megadja, hogy egyetlen vonalállapot hány bit információt hordoz.

Az ~ szerint az adatvonalak lehetnek: 1) kis sebességű (keskenysávú) vonalak, 300 bit/sec alatti, 2) középsebességű (hangfrekvenciás) vonalak, 300-9600 bit/sec, 3) nagy sebességű (szélessávú) vonalak, 9600 bit/sec feletti ~gel. Igen gyakori a 48 kbit/sec sebesség, de a felső határ egyre jobban kitolódik. Üzemelnek már 240 kbit/sec, sőt 0.5 Mbit/sec és 1 Mbit/sec sebességű vonalak is. A mikrohullámú, csőtápvonalas és lézersugaras összeköttetéseknél a fenti értékeknél több nagyságrenddel nagyobb ~ is elérhető.

adatátviteli szoftver (communication software) – az →operációs rendszerek kiegészítője: olyan speciális programcsomag, amelynek feladata az →adatátviteli hálózat vezérlése, valamint az →üzenetkapcsolás v. →csomagkapcsolás útján létrejövő információáramlás illesztése az operációs rendszer egészéhez. Az üzenetek továbbítása mindig valamilyen vezérlési eljárás szerint történik, adatátviteli vezérlő és azonosító karakterekkel folytatott párbeszéd útján. E folyamat vezérlése, valamint az üzenetek (csomagok) felépítése, szétválasztása, átalakítása, sorba állítása, adatvonalon történő továbbítása, az üzenetek esetleges kódolása- dekódolása stb. az ~ legfontosabb feladatai közé tartozik.

adatbank (data bank) – nagyobb összefüggő terület (tudomány, szakma, vállalat stb.) összes adatának bizonyos feladat v. feladatcsoport szempontjából tárolt rendszere. Az esetek többségében az ~ csak információs célokat szolgál, s egyszerre nagyszámú felhasználó rendelkezésére áll. Alapvetően abban különbözik az →adatbázistól, hogy az adatok szervezése az →adathordozókon messzemenően figyelembe veszi a felhasználók igényeit.

adatbázis (data base) – olyan, egymással kapcsolatban álló adatok összessége, amelyeket különféle felhasználók különböző csoportosításban használhatnak. Az adatok fizikai elhelyezése központilag, minimális redundanciával történik. Ugyancsak központilag ellenőrzött az adatok védelme, az új adatok bevitele és a meglévők módosítása v. törlése. Az →adatbázis-kezelő rendszer lehetővé teszi az arra jogosult felhasználóknak az ~hoz való hozzáférést úgy, hogy a felhasználói programok az adatok tárolási módjától függetlenek legyenek. Az ~ az adatok tárolási szervezésében elsősorban az →adatfüggetlenséget tartja szem előtt, s célja a lehető legtöbb alkalmazás optimális kielégítése (szemben az →adatbankkal, amely az adott felhasználók speciális igényeit veszi figyelembe az adatok szervezésében). Az ~-szervezés azon az elven alapszik, hogy egy adott logikai szervezés több, különböző fizikai szervezéssel is megoldható, s így lehetővé teszi a többszintű

adatfelfogást: másképp látja ugyanazokat az adatokat a →terminál képernyőjénél ülő felhasználó, másképpen az egyedi feldolgozói programtermékeket előállító szervező, megint másképpen az →adatbázis-felügyelő, s végül másképpen az adatok fizikai szervezését megvalósító rendszerprogramozó. Az ~ alapvető jelentősége, hogy megengedi bármelyik adatfelfogás megváltoztatását, miközben a többit változatlanul hagyja.

adatbázis-alséma (data-base subschema) – az →adatbázissémának olyan vetülete, amely egy meghatározott program által kezelni kívánt adatrészhalmaz logikai szerkezetének felel meg. Az ~ mindig származtatható kell hogy legyen az adatbázishoz tartozó egyetlen adatbázissémából. A nagyobb rugalmasság kedvéért az adatbázisséma és az ~ bizonyos jellemzőkben el is térhet egymástól. Az ~ban meghatározott kapcsolatokat az →adatbázis-kezelő rendszer alakítja át a sémának megfelelő formába, és a szükséges konverziókat is elvégzi. Az átalakítás folyamata, az adatbázisnak az ~val nem definiált része transzparens, rejtett az egyedi felhasználó részére. Így biztosítható az →adatfüggetlenség: az adatbázis globális logikai szerkezetében eszközölt változtatások nem vonják szükségszerűen maguk után a felhasználói programok módosítását is.

adatbázis-felügyelő (data-base administrator) – az a személy, aki az →adatbázissal kapcsolatos központi tevékenységekért felel. Fő feladatai: az adatbázis megszervezése (adatmodellek leírása, tárolási módjának meghatározása, keresési stratégiák megválasztása, hozzáférési jogok meghatározása); az adatbázis használatának állandó figyelése és a hatásfok jelentősebb csökkenésekor az adatbázis újraszervezése; az adatbázisban tárolt információk megfelelő védelme; az adatbázisban tárolt adatok helyességének állandó ellenőrzése; az újabb felhasználók igényeinek kielégítése stb. Az ~ egyik igen fontos feladata az adatbázis hangolása: ez azokat a – fizikai v. logikai szervezési szinten végzett – igazításokat, módosításokat jelenti, amelyek az adatbázis teljesítményének növelését szolgálják. Az ~ ugyanis a globális logikai leíráson (→adatbázissémán) keresztül látja az összes tárolt adatot, s az adatfelfogásnak ezen a szintjén beavatkozhat akár a logikai, akár a fizikai szervezésbe anélkül, hogy ez szükségessé tenné a felhasználói programok megváltoztatását.

adatbázis-hálózat (data-base network) – a területileg osztott adatkezelésnek az a speciális megvalósítása, amelynél egy globális logikai leírással (→adatbázissémával) rendelkező →adatbázis területileg elkülönített fizikai adatbázis-részhalmazokban testesül meg. Ezek a rendszerek az adatbázis-kezelés jellegzetes műveleteit és a táv-adatfeldolgozás műveleteit egyaránt megvalósítják.

adatbázis-hangolás (data-base tuning) – a már működő →adatbázis utólagos változtatása, átszervezése az egész rendszer teljesítményének szinten tartása v. növelése érdekében. Az ilyen típusú programtermékek →rugalmasságának hangsúlyozott jelentősége van, mert az utólagos alkalmazások speciális igényeit lehetetlen már az adatbázis tervezésénél figyelembe venni. Rendszerint a →fizikai adatbázis átszervezését jelenti, s ennek biztosítéka a tervezésnél már messzemenően szem előtt tartott fizikai és logikai →adatfüggetlenség. Sokszor az →adatvándorlás biztosítása is hangolási tevékenységet jelent.

adatbázis-kezelő rendszer (data-base management system) – azoknak a programoknak az összessége, amelyek – a felhasználói utasítások alapján – végrehajtják az →adatbázisban tárolt adatokra vonatkozó műveleteket. Elsődleges feladat az alkalmazói programok kiszolgálása úgy, hogy egyben biztosítsa az →adatfüggetlenséget, ill. az adatbázis védelmét is. A felhasználók csak az ~ közvetítésével férhetnek hozzá az adatbázisban tárolt információkhoz. Egyes rendszerek az →adatátviteli műveleteket is elvégzik.

adatbázis-másolat (data-base copy) – a fizikai →adatbázisról időszakonként automatikusan (az →adatbázis-kezelő rendszer funkciójaként) v. a felhasználói program által készített pontos másolat, amely az eredeti fizikai adatbázis meghibásodásakor a helyreállítás alapja (→előregörgetés, →visszagörgetés).

adatbázisnyelv (data-base language) – →adatbázisokban tárolt adatok leírására, kezelésére (előállítás, törlés, módosítások stb.), ill. lekérdezésére használt, jellegzetes követelményeknek eleget tevő magas szintű

→programozási nyelv (tkp. egy egész nyelvcsalád). Felépítését tekintve lehet olyan önálló nyelv, amelynek adat- és vezérlőszerkezetei a specifikus követelmények kielégítését támogatják, és lehet egy →befogadó nyelvként használt klasszikus programozási nyelv (pl. →COBOL, →PL/1, →FORTRAN, →Pascal stb.), amelybe az új struktúrákat utólag építették be. Az ~ három fő típusa: az →adatleíró nyelv, az →adatkezelő nyelv és a →lekérdező nyelv.

adatbázisséma (data-base schema) – az →adatbázis globális logikai szerkezetének leírása, a teljes logikai adatbázis blokkvázlata. Minden adatbázishoz egyetlen ~ tartozik az adatszervezés olyan logikai leírásával, amelynek révén az →adatbázis-felügyelő teljes egészében áttekintheti az adathalmazt. Az ~ból több különböző →adatbázis-alséma származtatható, s a felhasználónak csak ezeket kell figyelembe vennie. Az adatbázis-felügyelőnek biztosítania kell, hogy az egyes felhasználók által használt alsémák mind megfeleljenek az adott ~nak.

adatbázis-visszaállítás (data-base recovery) – hibás adatok bevitele v. helytelen kezelés következtében megsérült →adatbázis integritásának a visszaállítása; az →adatbázis-felügyelő alapvető feladata (→előregörgetés, →visszagörgetés).

adatbemutató rétegprotokoll (presentation layer protocol) – →hétréteges referenciamodell

adatbiztonság (data security) – az →adatbázisok (v. általában a tárolt információk) védelme a különböző negatív behatásoktól, mint pl. a fizikai meghibásodások, a feldolgozó programok hibái, a helytelen gépkezelői eljárások, az illetéktelen hozzáférések stb. Tágabb értelemben magában foglalja az →adatintegritást és az →adattitkosságot is.

adatblokk (data block) – programoknak az az egysége, amely egy helyre összpontosítja a programban használt összes adatot v. azok egy részét. Szövegszerűen az ~ definiálja mindazokat az azonosítókat, amelyeket a programban felhasznált változókhoz rendelnek, valamint azok összes attribútumát. Ezek a memóriakijelöléshez, ill. a rajtuk végzett műveletek típusának a meghatározásához szükségesek (pl. az →adatleíró főrész a →COBOL programozási nyelvben). Különleges jelentősége van az →újraindítható programok megvalósításában.

adatcsatorna (input-output channel) – →csatorna

adatcsomag (data packet) – az →adatátviteli hálózatban jól meghatározott formában kezelt adatblokk, amely a hasznos információ mellett magában foglalja az átvitelt vezérlő fejrészt is. Maximális mérete meghatározott, ennél hosszabb →üzeneteket több különálló ~ként kell továbbítani.

adatcsomag-egymásbaágyazás (data packet interleaving) – a →csomagkapcsolásos szervezésű →adatátviteli hálózatokban az adatcsomagok átvitelére használt módszer. A kapcsolóhoz →multiplexeléses üzemmódban nagyszámú egyszerű adatállomás csatlakozik egyetlen vonalon, de a tulajdonképpeni hálózat összeköttetései teljes adatcsomagokat továbbítanak. A különböző helyről érkező egyedi csomagokat a kapcsolónál építik fel, majd teljes csomagokat adnak az átviteli vonalra.

adatelem (data element) – olyan egységnyi adat, amelyre a programokban név szerint lehet hivatkozni.

adatelérési technika (data access method) – a külső →adathordozókon rögzített adatok elérésére kifejlesztett módszer. A legtöbb →operációs rendszer adatkezelő szubrutinjai két alapvető ~t biztosítanak: alaptechnikát és sorbaállító technikát.

Az alaptechnika csak →fizikai bejegyzésekkel dolgozik, →blokkoláskor a fizikai bejegyzések felépítéséről, ill. tördeléséről a felhasználónak kell gondoskodnia, csakúgy, mint a →b/k műveletek és a feldolgozó program szinkronizálásáról. Az alaptechnika legfőbb erénye, hogy minden szervezési módnál felhasználható.

A sorbaállító technika biztosítja a kényelmesebb feldolgozást: a rendszer szinte mindent automatikusan végez. A felhasználónak csak →logikai rekordokban kell gondolkoznia, a fizikai bejegyzések felépítését, ill. lebontását az operációs rendszer végzi. Ugyancsak a rendszer gondoskodik a b/k műveletek és a feldolgozás szinkronizálásáról. A sorbaállító technika csak szekvenciális feldolgozáskor használható.

Egyes operációs rendszerek kiterjesztették e két alapvető technikát a távolsági adatfeldolgozásra is (→alap-távfeldolgozási mód, →soros távfeldolgozási mód, →távfeldolgozási mód).

adat-előkészítés (data preparation) – a számítógéppel feldolgozható adatoknak a bevitelre alkalmas átalakítását és valamely →adathordozóra rögzítését magában foglaló tevékenység. Klasszikus formája az eredeti bizonylatokon összegyűjtött adatoknak az átmásolása →lyukkártyára, →lyukszalagra v. közvetlenül →mágneslemezre, →mágnesszalagra stb. Sokkal előnyösebb azonban az eredeti bizonylatok közvetlen olvasása, esetleg teljes kiiktatása úgy, hogy az adatokat közvetlenül a keletkezési helyen megfelelő →terminálokon keresztül küldik a feldolgozási helyre (→táv-adatfeldolgozás).

adatérvényesítés (data validation) – a hiteles és megbízható →adatfeldolgozásnak elengedhetetlen előkészítő művelete. Célja az adatok logikai helyességének, az általuk képviselt →információkhoz való viszonyuknak, az előírt értékintervallumokhoz tartozásuknak stb. ellenőrzése. Lényegében minden olyan ellenőrzési művelet, amelynek során a rendszerbe jutó adatok jellemzőit összevetik az adatleírásokban szereplő →attribútumokkal (→[adat]típus, mezőszélesség, értékhatárok, ábrázolási mód stb.).

adatfeldolgozás (data processing) – az →információnak jól meghatározott cél érdekében végzett elemzését, szintézisét, rögzítését, megváltoztatását, átvitelét stb. magában foglaló komplex tevékenység. Az ~ban az információk helyett az adatokkal végzik el a kívánt műveleteket: a gépi feldolgozás szempontjából szükségtelen az információk jelentésének ismerete, csak az a fontos, hogy az információkat tartalmazó adatokat egyértelműen azonosítani lehessen az eredetükkel.

adatfeldolgozó gép (data processing machine) – a legáltalánosabb értelemben minden olyan gép, amely digitális v. analóg jelek (adatok) tárolására, kezelésére (olvasás, írás, másolás, azonosítás, összehasonlítás, rendezés, számítás stb.) és továbbítására alkalmas.

adatfertőzés (data contamination) – valamely számítógéprendszerben tárolt adatok olyan jellegű módosítása, amely ellenkezik az →adatintegritás követelményeivel. Természetét tekintve lehet véletlenszerű, ha valamelyik részegység és/vagy a gépkezelő hibájából történik, és lehet rosszindulatú beavatkozás következménye (→vírus).

adatfüggetlenség (data independence) – az →adatbázisok egyik meghatározó jellemzője: az adatok szervezése és rögzítése úgy, hogy azok függetlenek legyenek a felhasználói programoktól. Ennek megvalósításához külön kell választani az adatok fizikai leírását és rögzítését a programok által látott logikai adatszerkezetektől. Az ~nek két szintje különböztethető meg.

A logikai ~ azt jelenti, hogy az adatok általános logikai szerkezetének (lényegében az →adatbázissémának) megváltoztatása nem teszi szükségessé a felhasználói programok módosítását.

A fizikai ~ azt jelenti, hogy az adatokkal kapcsolatos fizikai (tárolási és hozzáférési) jellemzőkben bekövetkezett változások nem teszik szükségessé az adatok általános struktúrájának, valamint a felhasználói programoknak a módosítását.

adatgyűjtés (data acquisition) – a →táv-adatfeldolgozásban használatos módszer, amelynél a földrajzilag távoli →terminálok (esetleg előkészített) adatokat közölnek a központi számítógéppel feldolgozás céljából. Az adatátvitel a konkrét alkalmazástól függően mind az egyes terminálok, mind a központi számítógép kezdeményezésére megindulhat. Az ~ lehet folyamatos, de lehetséges az is, hogy az adatokat a terminál helyén összegyűjtik valamilyen mágneses →adathordozón, és csak nagyobb egységenként küldik feldolgozásra.

adatgyűjtő rendszer (data acquisition system) – azok a berendezések és programok, amelyek a feldolgozó számítógéptől távol keletkezett adatok összegyűjtésére és a feldolgozás számára alkalmas formára alakítására szolgálnak. A begyűjtött információ →valós idejű feldolgozásnál azonnal feldolgozható, →kötegelt feldolgozásnál pedig tárolható későbbi feldolgozásra.

adathálózat (data network) – speciális →adatátviteli hálózat, amely elsősorban a számítógépes →adatfeldolgozásból származó információforgalom lebonyolítására szol-

gál (megkülönböztetésül a hang és kép átvitelére használatos hálózatoktól).

adathierarchia (data hierarchy) – a számítógépes →adatfeldolgozásban szereplő →adatok olyan ábrázolása, amely visszatükrözi a valós világból származó →információk között fennálló alá-, ill. fölérendeltséget. Egy adott szinten szereplő adat rendszerint az alsóbb szinteken szereplő adatok egész sorát képviselheti.

adathordozó (data media) – olyan fizikai közeg egy számítógéprendszerben, amely információ tárolására, ill. átvitelére alkalmas. Alapvető jellemzői: az adatok rögzítésének módja, az átvitel sebessége, az ~ cserélhetősége, a tárolt adatok elérhetőségének módszerei, az adatsűrűség stb. Az ~k több szempont szerint is csoportosíthatók:

Az adatok rögzítésére használt fizikai jelenség szerint lehetnek mechanikai ~k (pl. →lyukszalag, →lyukkártya), mágneses ~k (pl. →mágneslemez, →mágnesszalag stb.), optikai ~k stb.

A tárolt adatok elérhetősége szerint az ~k lehetnek soros elérésűek, ha az információkhoz csak a felvitel sorrendjében lehet hozzáférni (ilyenek pl. a lyukszalag, a lyukkártya, a mágnesszalag stb.), és lehetnek közvetlen elérésűek, ha minden →fizikai bejegyzésnek egyértelműen meghatározható fizikai címe van az ~n (pl. a mágneslemez, az optikai lemez stb.).

Az ~k lehetnek egyszeri felhasználásúak, ha a korábban tárolt információk többé nem törölhetők az ~ról (pl. lyukszalag, lyukkártya), s lehetnek újrafelhasználhatók, ha a tárolt információk törlése után újabb információk rögzíthetők ugyanazon a helyen (pl. mágneslemez, mágnesszalag stb.).

adathordozói címke (volume label) – a mágnesszalagtekercs, ill. a mágneslemezcsomag egyértelmű azonosítására szolgáló információk összessége. Szabványosított formája 80 →bájt hosszúságú blokk, amelynek első négy karaktere szolgál a címke azonosítására (VOL+sorszám). A VOL1 jelű ~ tartalmazza az adathordozó azonosítóját, a felhasználó tulajdonos nevét, a hozzáférés természetét (szabad v. jelszó ismeretét feltételező hozzáférés). Meghatározó jelentősége van az →automatikus kötetfelismerésnél.

adatintegritás (data integrity) – az →adatbázisokban szereplő adatok épségére, hitelességére, jó minőségére vonatkozó fogalom. Magában foglalja az →adatbázis-kezelő rendszernek azt a funkcióját is, amelynek célja a valamilyen okból meghibásodott adatbázis tartalmának visszaállítása a hiba előtti állapotra (→előregörgetés, →visszagörgetés).

adatkapcsolat (data link) – →adatátviteli hálózatok két csomópontja között időlegesen létrehozott kommunikációs csatorna, amely látszatra úgy viselkedik, mint az adott csomópontokat esetleg összekötő fizikai vezeték (vagyis az adatok a forrás által kiadott sorrendben érkeznek a vevőhöz). Az ~ tkp. több különböző hosszúságú és természetű vonalszakasz időleges egymáshoz való kapcsolásával hozható létre. A heterogén felépítés hajlamos a zajos átvitelre és az elkerülhetetlen késleltetésekre, ezért igen fontos a helyes átvitel biztosítását célzó vezérlés (→adatkapcsolati rétegprotokoll).

adatkapcsolati rétegprotokoll (data link protocol) – →adatátviteli hálózatokban használt →protokoll, amelynek elsőrendű feladata a zajos →adatkapcsolatoknak hibamentes kommunikációs csatornákra való átalakítása. Lényege, hogy az adatkapcsolatot felépítő vonalszakaszokon addig ismételteti az adatátvitelt, amíg hibátlan átvitelt nem biztosít a forrás és a vevő között (→hétrétegés referenciamodell).

adatkezelő nyelv (data manipulation language) – az →adatbázisokon végezhető műveletek leírására használt nyelv. Ezek a műveletek az adatbázis előállítására, módosítására v. lekérdezésére vonatkoznak, és módot adnak az adatbázis megnyitására és lezárására, a benne lévő →logikai rekordok módosítására és kiírására, új rekordok beszúrására stb. Az ~ lehet önálló nyelv, de beépülhet valamilyen magas szintű programozási nyelvbe is (pl. →COBOL, →PL/1, →FORTRAN stb.). Ha az ~ egy klasszikus programozási nyelv kiterjesztése, akkor független bármely egyedi →adatbázis-kezelő rendszertől. Ha viszont önálló nyelv, akkor független bármilyen programozási nyelvtől, s így sokkal nagyobb hatékonyságot biztosíthat, ugyanis nem szükséges, hogy →szintaxisa kompatibilis legyen valamely befogadó nyelv szintaxisával.

adatkoncentrátor (data concentrator) – speciális rendeltetésű, az →üzenetkapcsolás elvén működő kisszámítógép, amely a viszonylag lassú →adatvégállomásoktól származó üzeneteket összegyűjti, majd a fővonalon maximális sebességgel továbbítja a számítógépes központhoz. A feldolgozó központból sorosan, nagy sebességgel érkező üzeneteket fogadja, átmenetileg tárolja, majd szétosztja rendeltetési helyük szerint. Forgalomkiegyenlítő szerepe mellett az ~ olyan melléktevékenységeket is elvégezhet, mint az átalakítási műveletek, a hibafelderítés, a helyi adatállomások felügyelete stb.

adatkonverzió (data conversion) – adatok átalakítása egyik ábrázolási formából másikba. Az ~ érintheti az ábrázolásra használt fizikai jelenséget (→analóg-digitális konverter, →kártyaszalag konverter stb.), az ábrázolásra használt kódot (→decimális-hexadecimális konverzió, →decimális-oktális konverzió stb.), és érintheti az ábrázolási mód belső szerkezetét (→fixpontos számábrázolás, →lebegőpontos számábrázolás).

Különleges jelentősége van a programozás szempontjából az olyan ~nak, amely lehetővé teszi különböző →[adat]típusoknak a felhasználását vegyes kifejezésekben. →Gépi nyelvű v. →asszembler nyelvű programozásnál a programozó feladata az ~ tényleges elvégzése, a vegyes kifejezéseket megengedő magas szintű programozási nyelvek (pl. →FORTRAN) →fordítóprogramjai viszont automatikusan beépítik a →tárgyprogramba a szükséges ~kat megvalósító →szubrutinokat.

adatkonzisztencia (data consistency) – számítógépes feldolgozásban részt vevő adatok alapvető tulajdonsága, amely azt jelenti, hogy bizonyos adat bármilyen átalakításon esik is keresztül a program végrehajtása közben, mindig hűen visszatükrözi azt az objektív valóságot, amelyre vonatkozik. A klasszikus programozási nyelvek jóformán nem nyújtottak semmi segítséget az ~ biztosításában. Az →erősen típusos nyelvek megjelenésével jelentkezett nyelvi szinten is ez a követelmény: az →[adat]típusok (sőt az →Adában a →származtatott [adat]típus) biztosítják bizonyos fokig, hogy a belső ábrázolástól függetlenül az adatok csak olyan változásokon mehessenek keresztül (ill. olyan műveletekben vehessenek részt), amelyek nem sértik meg az ~ szabályait.

adatláncolás (data chaining) – →csatornaprogramokban használható láncolási módszer: több azonos típusú →csatornaparancsot jelent, amelyeknek különbözőek a tárolási tartományaik. ~nál az →adatcsatorna a soron következő parancsot csak arra használja, hogy más tárolási zónát jelöljön ki az eredeti bemeneti, ill. kimeneti művelethez, miközben a →periferikus mű ugyanazt a műveletet folytatja.

adatleíró főrész (data division) – →COBOL programoknak az a része, amelyben a program által feldolgozandó adatok tulajdonságait (típus, ábrázolási mód, sablon, struktúra stb.) írják le. Több részre tagozódik: FILE SECTION – az →[adat]állományok leírása; WORKING-STORAGE SECTION – a munkarekeszek leírása; LINKAGE-SECTION – a →formális paraméterek leírása; REPORT SECTION – a →táblázatgenerátor által kezelt rekordok leírása.

adatleíró nyelv (data description language) – az →adatbázisokban tárolt adatok, ill. a közöttük fennálló logikai kapcsolatok leírására használt nyelv. Alapvető eszköze az →adatfüggetlenség technikai megvalósításának, ugyanis az ~ segítségével írhatók le az adatbázist jellemző →adatbázisséma és az →adatbázis-alsémák.

Az adatok leírását három különböző szinten lehet elvégezni:

1) A felhasználó szintjén, ami az adatbázis egyik alsémáját jelenti. A megfelelő ~ lehet önálló nyelv, de igen gyakran egybeesik a felhasználói program nyelvével (ez rendszerint →COBOL).

2) Az adatok globális logikai kapcsolatának szintjén, amely a teljes adatbázisséma leírását tartalmazza. A megfelelő ~ rendszerint önálló nyelv.

3) Az adatok fizikai tárolásának szintjén, amely az egyes adatok fizikai elérhetőségét, tárolási módját és a közöttük fennálló fizikai kapcsolatokat írja le. Ezt a leírást is rendszerint önálló nyelv segítségével valósítják meg.

adatmodell (data model) – valamely természeti, technikai, gazdasági, társadalmi stb. jelenség mennyiségi jellemzőinek olyan strukturált együttese, amely figyelembe veszi a számítógépes adatfeldolgozás által támasz-

tott követelményeket is (elsősorban az →adatkonzisztenciát).

adatmodul (data module) – a →moduláris programozás elvei alapján előállított program olyan passzív →modulja, amely egy helyre összpontosítja a program által használt öszszes adatmezőt. Jelenléte a programokban több lényeges előnnyel is jár a programfejlesztés során.

Az aktív modulokat úgy is elő lehet állítani, hogy mindegyik tartalmazza saját adatmezőit, s →külső hivatkozás útján elérhessék egymás adatmezőit is. Amikor ezekből a modulokból felépítik a teljes programot, könnyen megtörténhet, hogy a rendelkezésre álló →operatív memória kevésnek bizonyul, s így →átfedési szerkezetet kell előállítani. Az átalakítás nagy nehézségekbe ütközhet, ha két modul, amely hivatkozik egymás adatmezőire, nem lehet egyszerre a memóriában. Ha viszont az összes ilyen közösen használt adatmezőt egyetlen ~ban helyezik el, s ez a rezidens részben (→gyökérszegmensben) kap helyet, az átfedési szerkezet könnyen megvalósítható.

Az ~ használata rendkívül előnyös a tesztelés szempontjából is. A különböző tesztmenetekben más és más ~t lehet a programba illeszteni, a teljes program újrafordítása nélkül. Az aktív modulok lefordított, →áthelyezhető bináris formában kapcsolhatók a módosított és újrafordított ~lal. A programok élesfutási karbantartását is könnyebbé teszi az ~, mert esetleges javításkor az adatszerkezet egy helyen, könnyen áttekinthetően módosítható, és nem kell végigjavítani az összes modult, amelyeket a módosítás érint.

adatnév (data name) – programozási objektumok azonosítására szolgáló →alfanumerikus karakterlánc. Képzésére a konkrét →programozási nyelv, ill. a hozzá tartozó →fordítóprogram ad pontos szabályokat.

adatsín (data bus) – →sínrendszer

adatstruktúra (data structure) – →adatelemek olyan rendezett halmaza, amelyben a felépítő elemek között szerkezeti összefüggések állnak fenn. Az ~t jellemző információk:

1) a hozzáférés szempontjából az adatelemek között milyen sorrendi összefüggések léteznek;

2) milyen eljárások útján lehet az egyes adatelemeket elérni;

3) hogyan lehet az ~t felépíteni és esetleg megváltoztatni.

Ha az ~hoz tartozó adatelemek azonos típusúak, akkor homogén, ellenkező esetben heterogén ~ról beszélnek.

adatszegmens (data segment) – olyan →szegmens, amely csak – más (program)szegmensekből olvasható és/vagy felülírható – adatokat tartalmaz. Különleges jelentősége van az →újraindítható programok megvalósításában.

adatszótár (data dictionary) – programtermékek tervezésében és dokumentálásában használatos segédeszköz, amely táblázatszerűen gyűjti egybe a feldolgozandó adatok szintaktikai és szemantikai jellemzőit. Egyértelmű összefüggést állít fel az adatok elnevezése és a hozzájuk rendelt programváltozók azonosítói között, meghatározza azok ábrázolási módját az →operatív memóriában és/vagy a külső →adathordozókon, megjelöli az adatok forrását és a helyességükkel kapcsolatos elvárásokat stb. Alapvető jelentősége van a rendszerelemzők, -tervezők és -programozók közötti információcsere redundanciamentes, pontos megvalósításában, a programtermékek dokumentálásában és későbbi karbantartásában.

[adat]típus ([data] type) – azoknak az értékeknek a rendezett halmaza, amelyeket egy →változó v. egy kifejezés felvehet. Az ~ az →adatabsztrakció egyenes következménye a programozási nyelvek elméletében és gyakorlatában. Legfontosabb jellegzetességei:

1) minden számításba jövő érték egy és csakis egy ~osztályba tartozik;

2) minden ~hoz jól meghatározott művelethalmaz tartozik;

3) minden művelet meghatározott ~ú érték(ek)en működik, és meghatározott, rendszerint az előzővel azonos ~ú értéket szolgáltat;

4) adott változóhoz az ~ explicit v. implicit →deklaráció útján kapcsolódik;

5) minden változóhoz tartozik egy bizonyos ~, és egy változóhoz csakis egy ~ tartozhat;

6) az ~ra vonatkozó információk a magas szintű programozási nyelvekben egyrészt arra szolgálnak, hogy megakadályozzák, ill. felfedjék a program hibás szerkezeteit, másrészt pedig meghatározzák az adatok ábrá-

zolásának módját és a rajtuk végzendő műveleteket (→erősen típusos nyelv, →típusegyezés).

adattitkosság (data privacy) – az →adatbázisokban v. →[adat]állományokban tárolt adatok illetéktelen felhasználás elleni védettsége. Magában foglalja az →adatbázis-kezelő rendszernek azt a funkcióját is, amely lehetővé teszi az →adatbázis-felügyelő számára, hogy maga döntse el, mikor, hogyan, kinek és milyen jellegű információk szolgáltathatók ki.

adattömörítési módszer (data concentration method) – az →adathordozók kihasználási fokának növelésére szolgáló módszer. Két alapvető típusa: a →logikai rekordok összevonása a →fizikai bejegyzésekben (→blokkolás), ill. a fizikai bejegyzésekben szereplő adatmezők tömörítése. Az utóbbi típushoz tartozó alábbi módszerek az egyes bejegyzések hosszát csökkentik le, információtartalmuk változatlan marad.

1) Ismétlődő karakterek elnyomása. Ha ugyanaz a karakter háromnál többször ismétlődik, akkor az egész lánc helyettesíthető két karakterrel: az első az ismétlési tényezőt, a második az ismétlődő karaktert jelöli (pl. a numerikus adatok vezető nullái v. a szöveges információk ismétlődő szóközei).

2) Üres területek kiiktatása változó hosszúságú →rekordok használatával.

3) Redundáns adatelemek kiiktatása, a különböző →[adat]állományokban található azonos adatelemek többszöri tárolásából eredő redundancia megszüntetése.

4) Tisztán numerikus adatok bináris ábrázolása.

5) Általánosan használt adatelemek helyettesítése. Pl. a keresztnevek 90%-a helyettesíthető egyetlen kóddal; szövegeknél a gyakori szavak v. kifejezések kódkarakterrel helyettesíthetők.

6) Megfelelő kódolás megválasztása. Pl. az ötbites Baudot-kód használata a nyolcbites →EBCDIC- v. →ASCII-kód helyett; változó hosszúságú karakterkódolás alkalmazása olyankor, amikor bizonyos karakterek előfordulási gyakorisága jóval meghaladja az összes többi karakter előfordulását (→Huffmann-kód).

Speciális ~eket használnak az →adatátviteli hálózatokban is az átviteli idő csökkentésére.

adatútképzés (routing) – →adatátviteli hálózatokban használt vezérlési algoritmus. Célja annak az összeköttetésnek a kijelölése, amelyen keresztül egy →adatcsomag v. →üzenet továbbításra kerül. A hálózat struktúrájához jól alkalmazkodó eljárás nagymértékben befolyásolja annak adatáramlási kapacitását, csökkenti az átlagos késleltetéseket. A legismertebb eljárások közé tartozik az →alkalmazkodó adatútképzés, a →címlista szerinti adatútképzés, a →heurisztikus adatútképzés, a →minimálsúlyozású adatútképzés, a →véletlenszerű adatútképzés, az →elárasztásos adatútképzés stb.

adatútképzési táblázat (routing table) – →adatátviteli hálózatokban a csomópontokhoz kapcsolódó táblázat. Célja annak a javasolt kimeneti összeköttetésnek a megállapítása minden egyes rendeltetési hely számára, amelyet az →adatcsomagnak v. →üzenetnek használnia kell.

adatvándorlás (data migration) – az érdeklődési szintnek megfelelő adattárolási módszer dinamikus változtatása. A különböző típusú →adathordozókat az elérési idejükben és árukban mutatkozó különbségek alapján többszintű rendszerekbe szervezik, s így megvalósítható, hogy a gyakran használt adatokat drágább, de gyorsan elérhető tárolón (pl. →mágneslemezen), a ritkábban igénybe vett adatokat lassúbb, de jóval olcsóbb tárolón (pl. →mágnesszalagon) rögzítsék. Bizonyos tárolt adatok iránt az érdeklődés időben változó lehet, ezért a megfelelő hatékonyság érdekében gondoskodni kell arról, hogy a tárolás helye és az érdeklődés foka mindig összhangban legyen egymással. Néhány →adatbázis-kezelő rendszerben az ~ automatikusan zajlik le, más rendszerekben viszont az →adatbázis-felügyelő kezdeményezi. Az ~ sokszor az adatbázis hangolásának része.

adatvégállomás (terminal equipment) – →adatátviteli hálózatokban megtalálható →adatállomás, amelyen keresztül megvalósul a távfeldolgozási rendszer és a távoli környezet kapcsolata. Vonalcsatlakozó segítségével illeszkedik az adatátviteli hálózathoz, periferikus berendezései a →terminálok. Ezek egy része olyan közel is elhelyezkedhet a számítógépes feldolgozó rendszerhez, hogy

esetleg közvetlenül is kapcsolódhat soros v. párhuzamos átvitelű illesztéssel.

adatvezérlő blokk (data control block) – valamely →[adat]állomány logikai szintű feldolgozásához szükséges olyan információk táblázata, amelyeket az →operációs rendszer adatkezelő szubrutinjai (→[adat]állomány-kezelő rendszer) használnak fel. Az ~ban rögzített legfontosabb adatok:

1) az →adathordozót jellemző típuskód és azonosító;

2) az [adat]állományt jellemző szervezési mód, a →logikai rekordok formátuma és hossza, a →fizikai bejegyzések mérete stb.;

3) a feldolgozási folyamatot jellemző →pufferek száma, felépítése és kezelési módja, az elérési mód stb.;

4) a különleges szolgáltatásokat jellemző adatok: a feldolgozás menete alatt adódó különleges helyzeteket megoldó feladatok leírása, szekvenciális feldolgozásnál az utolsó rekord beolvasása utáni teendő, a felhasználói címkék feldolgozása, javíthatatlan hibák kezelése stb.

Az ~ kitöltése általában több lépésben történik. Elsőként a →fordítóprogramok helyeznek el információkat a felhasználó leírása alapján. Második lépésben a →kapcsolóprogram ír az ~ba, az [adat]állományokra vonatkozó vezérlő utasítások alapján. Utoljára az [adat]állomány megnyitásakor kerülnek információk az ~ba a →szabványos címkékből.

AFIPS (American Federation of Information Processing Societies) – amerikai számítástechnikai szervezetek 1961-ben alapított szövetsége. Fő célkitűzése a szakterületen elért legfontosabb tudományos és technikai eredmények nemzetközi terjesztése. Fontos rendezvényei a kétévenkénti kongresszusok, amelyeknek alapvető tematikája a programozáselmélet és a számítógép-alkalmazás.

AHPL (A Hardware Programming Language) – magas szintű programozási nyelv, amelyet numerikus számítógépek működésének leírására fejlesztettek ki. Fontos eszköze a számítógépek számítógépes tervezésének.

akkumulátor (accumulator) – az →aritmetikai egységnek olyan (egy v. több szó hosszúságú) speciális →regisztere, amely az aritmetikai műveletek előtt rendszerint az első operandus, a művelet-végrehajtás után pedig az eredmény tárolására szolgál. Az ~ hosszúsága általában a többi munkaregiszter hosszával, ill. a →szó hosszúságával egyezik meg.

aktív [adat]állomány (active file) – olyan →logikai [adat]állomány, amely érvényes megnyitási műveleten esett keresztül, s ezáltal a felhasználó rendelkezésére áll szerkesztési, karbantartás v. feldolgozás jellegű műveletek elvégzésére.

aktív csillag (active star) – olyan →csillaghálózat, amelynél a központi csomópont feldolgozási műveleteket is végez a külső csomópontok közötti összeköttetés biztosítása mellett. Alapvető hátránya, hogy a központi csomópont hibája az egész hálózat tevékenységét hátrányosan befolyásolja.

aktív előfizető (active abonent) – az →adatátviteli hálózat szolgáltatásait adott időpontban ténylegesen igénybe vevő előfizető. Az adatátvitelt vezérlő →szoftver szempontjából fontos szervezési fogalom (adó-, ill. vevőcímzés, várakozások szervezése stb.).

aktivitás (activity) – az a megfelelően kódolt információ, amely arra utal, hogy egy-egy hasznos információegységre történt-e utalás adott időperiódus alatt, v. sem. →[Adat]állományoknál az ~ megmutatja, hogy mennyire gyakran hivatkozik a felhasználói program bizonyos rekordokra. Ennek ismeretében az →indexszekvenciális, ill. →direkt szervezésű [adat]állományt úgy lehet átszervezni, hogy a leggyakrabban lekérdezett rekordok az [adat]állomány főrészébe kerüljenek, javítva ezáltal a feldolgozás hatásfokát. Különleges jelentősége van a különböző →laphelyettesítési stratégiák megvalósításában, ugyanis az olyan lapokat, amelyekre a betöltés óta nem történt felülírásos hivatkozás, nem szükséges kimenteni a külső memóriába.

aktívlapregiszter (active page register) – →virtuális címzéssel működő rendszerek speciális regisztere. A →virtuális címeknek →fizikai címekre való leképzéséhez szükséges – a rendszerre jellemző – információkat tartalmazza.

aktív program (active program) – többfelhasználós környezetben (→multiprogramozás, →időosztásos üzemmód) az a program, amely adott pillanatban kizárólagosan használja a →központi vezérlőegységet. Egyprocesszoros rendszerben természetesen egyszerre csak egy felhasználónak lehet ~ja.

aktív számítógép (active computer) – tartalék →redundanciát megvalósító →többprocesszoros rendszerekben az a számítógép, amely adott pillanatban ténylegesen részt vesz az adatfeldolgozásban v. az →adatátviteli hálózat vezérlésében.

aktuális érték (actual value) – adott →programban szereplő adat pillanatnyi állapotára jellemző érték.

aktuális kulcs (actual key) – →direkt szervezésű [adat]állományok →logikai rekordjainak visszakeresésére használt éppen érvényes adatelem.

aktuális paraméter (actual parameter) – valamely →eljárás bemenő adatát képező, annak konkrét végrehajtásához szükséges tényleges számítási érték. Az eljárás →formális paramétereinek megfelelő ~ek előállítása és átadása a hívott eljárásnak, az →eljáráshívásnak meghatározó művelete. Az ~ek és a formális paraméterek közötti információátvitel három alapvető módszerrel történhet: →cím szerinti hivatkozással, →érték szerinti hivatkozással és →név szerinti hivatkozással.

aktualizálás (updating) – valamely →újrahasználható adathordozón rögzített információnak „naprakész" állapotba hozása. Alapvető jellegzetessége, hogy a →fizikai bejegyzéseknek csak a tartalmát érinti, nem befolyásolja a fizikai szerkezetet megvalósító járulékos adatokat, ill. a →logikai rekordok eredeti sorrendiségét (szemben a →karbantartással).

akusztikus csatoló (acoustic coupler) – speciális →modem, amely →interfészként működik számítógéprendszerek és a nyilvános telefonhálózat között.

akusztikus memória (acoustic memory) – az első generációs számítógépekre jellemző memóriatípus, amely az információt nyomáshullámok formájában tárolta. Újabb változatánál mágneses vékonyrétegeket használnak, amelyek hosszirányú mechanikus impulzusok hatására megváltoztatják mágneses jellemzőiket. Ezek az új ~k nagy mennyiségű információ (kb. 10^8 bit) tárolására alkalmasak, hozzáférési idejük mikroszekundum nagyságrendű, tehát könnyen felveszik a versenyt a mágneslemezes tárolóval.

alacsony szintű nyelv (low-level language) – →gépi nyelvek és →asszembler nyelvek összefoglaló megnevezése. Alapvető jellemzője az ~nek, hogy nagymértékben gépfüggő, más szóval: a felhasználható adat-, ill. vezérlőszerkezetek közvetlenül visszatükrözik a célszámítógép →architektúráját.

alap-adatelérési technika (basic access method) – alapvető →adatelérési technika, amelynek lényege az, hogy →fizikai bejegyzésekkel (blokkokkal) dolgozik. Amennyiben →blokkolást használnak, a blokkok tördelését, ill. felépítését a programozónak kell elvégeznie. Az ~ nem szinkronizálja a →b/k műveleteket és a feldolgozó programot. Előfordulhat tehát, hogy egy olvasási művelet még nem fejeződött be, amikor a program már fel akarja dolgozni annak eredményét, ezért külön szinkronizáló →makroutasításokat kell használni (ezek szükség esetén felfüggesztik a program végrehajtását a b/k művelet befejezéséig). Az ~ bármilyen adatszervezési módszernél használható, s ennek megfelelően a következő technikák ismeretesek: →szekvenciális alap-adatelérési technika, →particionált alap-adatelérési technika, →indexszekvenciális alap-adatelérési technika, →direkt alap-adatelérési technika, valamint a távfeldolgozásban az →alap-távfeldolgozási mód

alapértelmezési érték (default value) – valamely operandus v. →mező értékére vonatkozó előírás, ha a felhasználó nem adott rá konkrét meghatározást, értéket.

alapgép (base machine) – →operációs rendszer nélküli számítógép, amely kizárólag csak a →központi vezérlőegység által felismert utasításokat képes végrehajtani (szemben a →kiterjesztett géppel).

alaplap (mother board) – →személyi számítógépek alapvető szerkezeti egysége, áramköri lapja, amely a rendszer legfontosabb funkcionális elemeit hordozza. Itt kap helyet a →központi vezérlőegység szerepét játszó →mikroprocesszor, a →központi memória, az órajel-generátor, a →programozható megszakításvezérlő, a →programozható perifériaillesztő, esetleg az →aritmetikai társprocesszor stb. Az ~hoz dugaszolással további funkcionális elemeket hordozó →bővítőkártya is csatlakoztatható.

alapsávi jel (baseband signal) – →adatátviteli műveletekben az az eredeti jel, amellyel az átviteli hullámformát modulációval létrehozzák. A távbeszélő technikában ez a beszéd jelalakja.

alapsávú adatátvitel (baseband networking) – moduláció nélkül megvalósított, digitális adatátvitel, amelyhez sodrott érpárú v. koaxiális kábeleket használnak. A →helyi hálózatok jellegzetes üzemmódja.

alapszámos írásmód (radix notation) – számok olyan helyértékes ábrázolási módja, amelynél a szám értékét a számjegyek és a helyértékek szorzatának összege adja, a helyérték pedig az alapszám megfelelő hatványa.

alapszimbólum (basic symbol) – valamely →programozási nyelv →ábécéjéhez tartozó karakter, vagyis az illető nyelv →terminális szimbólumait felépítő betűk, számjegyek v. speciális karakterek egyike.

alap-távfeldolgozási mód (basic telecommunication access method) – táv-adatfeldolgozásban használható →alap-adatelérési technika. Elsősorban a szerényebb rendszerek kezelésére alkalmas, v. akkor, amikor a követelmények eltérnek a szokásostól. Alapvető szolgáltatásai: →terminálok címzése, lekérdezése, üzenetek fogadása, dinamikus →pufferek kezelése stb.

alárendelt processzor (slave processor) – számítógépes →architektúrában v. →számítógépes hálózatban található olyan →processzor, amely csak bizonyos célfeladatok elvégzésére szolgál, rendszervezérlési funkciókat nem láthat el. Ilyen vonatkozásban – pl. a →programozható megszakításvezérlő, a →programozható perifériaillesztő, az →aritmetikai társprocesszor stb., hálózatokban pedig a →kommunikációs processzor.

álcím (dummy address) – a →gépi utasítás címrészén található olyan bitvektor, amelyet a →központi vezérlőegység nem címként, hanem az illető utasítás elvégzésével kapcsolatos kódolt információként értelmez. Pl. a →léptetésnél a művelet típusát, módját, irányát, a lépések számát jelöli.

áleszköz (pseudo device) – olyan rendszerelem, amelyet a felhasználó →b/k műveletekre használ, bár az nem egy bizonyos fizikai eszköz. E műveletek tényleges végrehajtására az →operációs rendszernek el kell végeznie az ~ átirányítását (leképzését) egy meghatározott fizikai eszközre. Programozástechnikai jelentősége, hogy a fizikai eszközök konkrét jellemzőitől független programokat lehet előállítani.

alfanumerikus (alphanumerical) – az angol ábécé betűit, a decimális számjegyeket és bizonyos speciális szimbólumokat közösen minősítő attribútum.

alfanumerikus kijelzés (alphanumerical displaying) – számítógéprendszer által (→nyomtatón, képernyőn stb.) szolgáltatott, alfanumerikus karakterek formájában megjelenő információ.

alfanumerikus kód (alphanumerical code) – az angol ábécé 26 betűjének, a tíz decimális számjegynek és bizonyos számú speciális szimbólumnak a kódolására alkalmas →kód (leggyakrabban használt változatai az →ASCII-kód, ill. az →EBCDIC-kód).

alfanumerikus konstans (alphanumerical constant) – idézőjelekkel határolt karaktersorozat, amelyben a gépi karakterkészlet minden szimbóluma szerepelhet.

alfanumerikus terminál (alphanumerical terminal) – kizárólag alfanumerikus jelsorozatok vételére és/vagy közvetítésére alkalmas →terminál.

ALGOL (ALGOrithmic Language) – elsősorban tudományos-műszaki feladatokra orientált magas szintű →programozási nyelv. 1958-ban kezdték el kidolgozni, 1960-ban egy Párizsban tartott értekezleten fogadták el ALGOL-60 néven, majd 1968-ban hagyták jóvá kibővített változatát ALGOL-68 néven. Három alapvető formája ismeretes.

A hivatkozási forma a legáltalánosabb, nem tartalmaz előírásokat az adatátviteli műveletekre vonatkozóan.

A gyakorlati (implementációs) forma a konkrét számítógépeken megvalósított adatátviteli műveleteket is tartalmazza, de a hivatkozási nyelv egyszerűsített változata.

A publikációs forma a nemzetközi értekezés legfontosabb eszköze volt a szakterületen, ma már inkább a →Pascal szabványos formáját használják.

Az ~ legfontosabb tulajdonsága a moduláris, blokkokra épülő szerkezet, amely lehetővé teszi az →operatív memória dinamikus felhasználását, a →rekurzivitás programozhatóságát.

algoritmikus függvény (algorithmic function) – olyan függvény, amely →algoritmus segítségével megvalósítható. Más megközelítésben: létezik egy olyan algoritmus, amely a függvény értelmezési halmazának minden

elemére képes előállítani az illető argumentum(ok)ra érvényes függvényértéket.

algoritmikus nyelv (algorithmic language) – magas szintű programozási nyelv, amely megfelelő eszközökkel rendelkezik →algoritmusok leírására. Általánosan ismert ~ek: →ALGOL, →FORTRAN, →COBOL, →PL/1, →Pascal, →Ada stb.

algoritmus (algorithm) – véges számú, előre ismert műveletek olyan sorozata, amely megadja egy feladat v. egy problémakör megoldásának pontos leírását, a megoldáshoz vezető műveletek természetét és sorrendiségét. Három alapvető tulajdonsága: 1) az általános érvényűség (a feladatok egész osztályát képes megoldani bármilyen bemenő adatokra); 2) a determinizmus (az eljárás minden lépése előre ismert és minden műveletet előre ismert művelet követ); 3) a véges jelleg (a lépések száma és a végrehajtás ideje egyaránt véges).

Számítástechnikai értelmezésben a harmadik tulajdonság bizonyos megszorításoknak van alávetve. A matematikai analízisben számtalan olyan ~ van, amely a határérték fogalmát használja, tehát lényegében végtelen sok lépést követelne, ha abszolút pontosságra törekednének. Gyakorlati szempontból tehát csak a relatív pontosság lehet a célja egy ~nak. Egy, a matematikában „végesnek" ismert ~ a számítógépen elvesztheti ezt a tulajdonságát.

Ma már elméletileg is igazolt tény, hogy léteznek algoritmikusan megoldhatatlan feladatosztályok is, amelyek csak közelítő eljárásokkal oldhatók meg.

alkalmazkodó adatútképzés (adaptive routing) – olyan →adatútképzés, amelynél a rendszer alkalmazkodni képes a hálózat változásaihoz. Ezeket pl. a forgalmi viszonyok v. meghibásodások idézhetik elő. Az →adatcsomagok, ill. →üzenetek útját úgy kell kijelölni, hogy minimálisra csökkenjenek a helyi →torlódások, az összeköttetés- v. csomóponthibák megkerülésére pedig alternatív választásokra is lehetőség legyen. Az alkalmazkodáshoz szolgáló tapasztalat magából a lebonyolítandó adatforgalomból származik.

alkalmazkodóképesség (adaptability) – programozástechnikai szempontból programtermékeknek az a tulajdonsága, hogy minimális változtatást igényelnek, ha az üzemi feltételek (→hardver, →operációs rendszer, →fordítóprogram stb.) megváltoznak v. ha a programtermékek feladatait kell módosítani.

alkalmazói rétegprotokoll (application layer protocol) – →hétréteges referenciamodell

alkalmazói szoftver (application software) – általános értelmezésben: a felhasználók által előállított programtermékek összefoglaló megnevezése; egy konkrét számítógépre vonatkoztatva: az adott gépen megoldott feladatok programrendszereinek összessége (megkülönböztetésül a számítógéppel együtt szállított →rendszerszoftvertől). Kereskedelemben megvásárolható szoftvercsomag.

állandó virtuális összeköttetés (permanent virtual circuit) – →csomagkapcsolásos →adatátviteli hálózatokban két →adatvégállomást úgy összekötő →kommunikációs csatorna, hogy az állandó összeköttetés látszatát nyújtsa a felhasználónak (→bérelt vonal).

állapotdiagram (state diagram) – valamely →automata állapotainak ábrázolására használt →irányított gráf. Csomópontjai az automata állapotait, élei a leképzési függvényeket jelképezik.

állapotegyenlet (next state equation) – az a logikai kifejezés, amely a bemeneti változók és az aktuális állapot függvényében meghatározza a szekvenciális logikai hálózat következő állapotát.

állapotinformáció (status information) – valamely →hardverberendezés v. →program állapotát jellemző, a számítógéprendszer szempontjából fontos – rendszerint bitvektor formájában kódolt – információ (→programállapotszó, →csatornaállapot-szó stb.).

állapotregiszter (status register) – speciális rendeltetésű →regiszter, amely bizonyos logikai v. fizikai →erőforrás állapotára vonatkozó információkat tartalmaz. A →megszakítási rendszerek alapvető eszköze.

állapotszó (status word) – olyan információvektor, amely egy folyamat végrehajtását jellemző paramétereket foglal magában, vagyis mindazokat az információkat, amelyekre a →központi vezérlőegységnek szüksége van a folyamat végrehajtásának megindításához v. →megszakítás utáni újraindításához. Az ~ komponensei az →operációs rendszer típusától függenek, de általában mindig megtalálhatók a következő információk: a folyamat

azonosítója, a hozzá tartozó →címtartomány, a következő végrehajtandó utasítás címe, a folyamat által használt regisztereknek a felfüggesztés pillanatában megmentett tartalma, a folyamat →prioritása, a védelmi kulcsok stb. Többfelhasználós környezetben minden folyamathoz saját ~ tartozik, amely alapvető eszköze az egyidejűleg végrehajtás alatt álló folyamatok összehangolt ütemezésének.

állomány [adat]típus (file [data] type) – →erősen típusos nyelvekben definiálható →strukturált [adat]típus, amely ugyanahhoz az [adat]típushoz tartozó, határozatlan számú adat sorozatára vonatkozik. Bár a →Pascal programozási nyelv mint új [adat]típust vezette be, lényegében a klasszikus →szekvenciális [adat]állománynak felel meg.

alprogram (subprogram) – →szubrutin

álszakasz (dummy section) – →asszembler nyelven kódolt programnak az a →szakasza, amely az →operatív memória egyik részének →címtartományát írja le. Az ~ban leírt azonosítók globális jellegűek, a programszakaszokból közvetlenül értelmezhetők.

általános célú számítógép (general-purpose computer) – olyan digitális számítógép, amely minden – →algoritmus segítségével leírható – feladat megoldására felhasználható (szemben a →célszámítógéppel, amely egy bizonyos feladat megoldására specializált).

általánosított információs rendszer (generalized information system) – olyan →információs rendszer, amely az adatfeldolgozást általánosított, előre megtervezett és paraméterezett eszközökkel végzi. A műveletek általánosítási szintje elég magas ahhoz, hogy ne függjön egy konkrét felhasználó – adatszerkezetekre és -kezelésre vonatkozó – igényeitől, de megfelelő paraméterek útján ki tudja szolgálni azokat.

általános regiszter (general register) – a →központi vezérlőegység olyan →regisztere, amely többféle funkciót is elláthat s a programokban használható mint pl. →bázisregiszter, →indexregiszter, →akkumulátor, részeredmények tárolására szolgáló gyorsmemória stb. Ha az ~t a program végrehajtása során különböző célokra használják, tartalmát az →operatív memóriában lehet tárolni.

alternatíva (alternative) – valamely vonatkozásban egyaránt lehetséges, de megvalósulásukban egymást kizáró cselekmények, megoldások v. választások, ill. ezek bármelyike.

alternatív adatútképzés (alternate routing) – →adatátviteli hálózatokban az →adatútképzésnek az a változata, amelynél másodlagos adatutakat is kijelölnek az adatforgalom számára (v. az esetleges vonalhibák kivédésére v. optimalizálási célokat követve).

áltetrád (pseudo decimal digit) – a decimális számjegyek bináris ábrázolásához legalább 4 →bitre (tetrád) van szükség, ezek viszont olyan kombinációkat is adnak, amelyekhez nincs rendelve decimális számjegy. A 16 lehetséges kombinációból csak 10-et használnak ki, így 6 ~ keletkezik. Ezeket bizonyos fokig hibaellenőrzésre használhatják, ugyanis megjelenésük azt mutatja, hogy eredetileg egy értékes jegy megváltozott.

alulcsordulás (underflow) – →lebegőpontos számábrázolásnál az az eset, amikor egy →aritmetikai művelet eredménye olyan kicsi, hogy az már nem ábrázolható a →karakterisztika számára fenntartott bitpozíciókon. Az ~ általában →csapdát jelent, s a program végrehajtása megszakad.

alulról felfelé haladó módszer (bottom-up method) – a →moduláris programozásban használható programfejlesztési stratégia. Elsősorban a kódolás és tesztelés fázisaiban előnyös, mivel a program létrehozását és bejáratását a legalsó logikai szinten kezdik, s a már kész, bizonyítottan helyes programegységekből építik fel a közvetlenül felette következő logikai szintet. Az ~ legfontosabb előnyei: a →modulok egymástól függetlenül kódolhatók és tesztelhetők; egy tesztmenet alatt egy modul különböző tesztadatokkal is vizsgálható; a kódolási és logikai hibák alacsony szinten, viszonylag könnyen felderíthetők.

Az ~ hátrányaként említhető, hogy a már tesztelt modulok kapcsolásánál esetleg előre nem látott logikai összeférhetetlenségek adódhatnak. Ez viszont elkerülhető, ha az ~rel megvalósított kódolást és tesztelést a →felülről lefelé haladó módszert használó rendszerelemzés és -tervezés előzi meg.

alulról felfelé haladó szintaktikai elemzés (bottom-up syntactical analysis) – →fordítóprogramok által a →szintaktikai elemzés megvalósítására használt stratégia. Lényege az, hogy a →szintaxisfát a →terminális szim-

bólumoktól (vagyis az adott mondat szimbólumaitól) kiindulva a gyökérelem felé haladva építi fel (szinonimája az összevonó szintaktikai elemzés).

álutasítás (dummy instruction) – →direktíva

analóg adatátvitel (analogue transmission) – →adatátviteli hálózatokban használt módszer a digitális információk átvitelére. Lényege, hogy az eredeti digitális jelet modulációnak vetik alá, s így az átvitelre →analóg csatornák is használhatók. Az →adatállomások →modemek segítségével csatlakoznak a vonalra. A vivőfrekvenciás rendszerek három alapvető modulációs eljárást alkalmaznak: amplitúdómodulációt, frekvenciamodulációt és fázismodulációt. Igen nagy sebességű adatátviteli rendszerekben a modulációs eljárások kombinációinak többszintes változatait is használhatják.

analóg-digitális konverter (analogue-to-digital converter) – olyan (legtöbbször elektronikus) berendezés, amely a bemeneten megjelenő →analóg jel értékét a kimeneten digitális jellé alakítja. Legfontosabb felhasználási területe a folytonos jelenségek numerikus számítógéprendszerekben való feldolgozása (pl. távmérés, távvezérlés, folyamatszabályozás stb.).

analóg jel (analogue signal) – valamely folytonos jelenséget jellemző mennyiség, amelynek mérőszáma valamilyen határok között bármilyen értéket felvehet.

analóg számítás (analogue computing) – matematikai műveletek végrehajtása →analóg számítógép segítségével. Rendszerint algebrai v. differenciálegyenlet-rendszerek megoldásának időben lefolyó változásait határozzák meg vele.

analóg számítógép (analogue computer) – olyan számítógép, amelyben a számítás eredményét valamely folytonos fizikai jelenség (legtöbbször áramerősség, feszültség, nyomás stb.) értékei adják. Az adott fizikai jelenséget leíró matematikai modell határozza meg, hogy az ~et milyen típusú feladat megoldására lehet használni. Alkalmazása akkor előnyös, ha mérési adatokat kell feldolgozni s az eredményként kapott jelekkel folytonos jellegű folyamatokat kell szabályozni. A →numerikus számítógéptől eltérően nem program alapján működik, s az előbbinél szokásos →memóriával sem rendelkezik. Az ~ számítási pontosságát az összetevő fizikai elemek pontossága határozza meg.

anonim változó (anonymous variable) – a →Prolog nyelvben használt olyan speciális →változó, amelyhez nem kapcsolódik →azonosító.

APL (A Programming Language) – magas szintű →programozási nyelv, elsősorban matematikai alkalmazások céljára. 1962-ben dolgozta ki K. E. Iverson amerikai matematikus, s rövid idő alatt a →BASIC mellett a legelterjedtebb →párbeszédes üzemmódú programozási nyelv lett. Alapvető tulajdonsága, hogy implicit módon kezeli a többdimenziós tömbökkel végzett műveleteket. Pl. két mátrix szorzására egyszerű utasítás létezik C ← A · B, nincs szükség a szorzás elemenként történő elvégzésére, mint az más programozási nyelveknél megszokott. Utólag módosított változatait sikeresen használják a →numerikus számítógépek működésének leírására. Legfontosabb gyakorlati megvalósítása az APL/360, amelyet az IBM számítógépeken implementáltak.

APLG (A Programming Language for Graphics) – az →APL programozási nyelv grafikus adatfeldolgozást is támogató változata.

Apple – →mikroszámítógépek és →személyi számítógépek előállításában élenjáró neves amerikai vállalat, amely LISA rendszerével először jelent meg a piacon teljesen grafikus →menüvezérelt üzemmódú →programozási környezettel. Ugyancsak ~-újítást jelentett a VisiCalc nevű programtermék, amely a →táblázatkezelők első képviselője. Macintosh számítógépe egyike a legelterjedtebb →munkaállomás típusú rendszereknek.

APT (Automatic Programming for Tools) – →FORTRAN típusú magas szintű →programozási nyelv, amelyet szerszámgépek vezérlésére szánt programok írására használnak. Az ~ nyelven pontok, egyenesek, körök, geometriai felületek stb. szabványos jelekkel, ill. szavakkal írhatók le. A program kezdetén az általános jellegű adatokat határozzák meg (pl. pontosság, amivel a görbéket egyenes szakaszokkal megközelítik), majd a megmunkálás során bejárandó út leírása következik.

architektúra (architecture) – számítástechnikai értelmezésben valamely számító-

géprendszer felépítésére vonatkozó fogalom, amely jellemzi az illető rendszerben található →erőforrások számát, természetét, a közöttük megvalósuló kölcsönhatásokat, a tervezésben és kivitelezésben felhasznált módszereket, elveket stb.

argumentum (argument) – a →paraméterrel egyenértékű kifejezés, ugyanis a programozástechnikai értelemben vett →eljárás függvénynek is tekinthető.

aritmetikai egység (arithmetical unit) – a számítógép →központi vezérlőegységének az a része, amely aritmetikai műveleteket (a négy alapműveletet, összehasonlítást, komplementumképzést, →léptetést stb.) végez →bináris számokkal v. binárisan kódolt decimális számokkal. Felépítését az operandusok ábrázolási formája határozza meg: eszerint az ~ lehet fixpontos, lebegőpontos v. decimális ~.

aritmetikai és logikai egység (arithmetical and logical unit) – →digitális számítógépek →központi vezérlőegységének az a része, amely →aritmetikai és →logikai műveleteket végez, a legtöbb esetben →fixpontos számábrázolású operandusokkal. A klasszikus →architektúrában rendszerint fixpontos ~ vesz részt a központi vezérlőegység felépítésében, a →lebegőpontos, ill. a decimális aritmetikai műveletek elvégzésére külön opcióként csatlakoztatnak megfelelő típusú →aritmetikai egységet (pl. ilyen az →aritmetikai társprocesszor is).

aritmetikai kifejezés (arithmetical expression) – magas szintű →programozási nyelvekben (→FORTRAN, →PL/1, →Pascal stb.) használt kifejezés, amely aritmetikai műveleti jelekkel (összeadás, kivonás, szorzás, osztás) kapcsolt aritmetikai operandusokból épül fel, esetleg zárójelekkel csoportosítva. Az aritmetikai operandusok lehetnek konstansok, egyszerű v. indexelt változók, standard függvényeljárások. →Szintaxisa a →BNF formalizmussal a következő módon írható le:

<aritm.kif> : : = <tag> | <aritm.kif>
<additív.op> <tag>

<tag> : : = <tényező> | <tag>
<multipl.op> <tényező>

<tényező: : = <operandus> | (<aritm.kif>)
| <tényező> <hatv.op> (<aritm.kif>)

<additív.op> : : = + | –

<multipl.op> : : = * | /

<hatv.op> : : = **

→ Szemantikáját az algebrai kifejezések kiértékelési sorrendjét meghatározó szabályok írják le, azzal a kiegészítéssel, hogy a soron következő művelet előtt esetleges átalakítások épülnek be, ha a két operandus nem azonos típusú (→adatkonverzió).

aritmetikai léptetés (arithmetical shift) – matematikai szempontból egy mennyiségnek a számrendszer alapszámával végzett szorzása v. osztása; gyakorlatilag egy →regiszterben tárolt bináris vektor bitpozícióinak megváltoztatását jelenti olyan módon, hogy az első bit (előjelbit) változatlan marad. A művelet végrehajtása a negatív számok ábrázolási módjától függ. →Komplemens kódú ábrázolásnál az egy pozícióval végzett ~ a következő módon történik: balra végzett ~nél a második bitpozíció tartalma elvész (nem léphet az előjel helyére), az utolsó bitpozíció pedig 0 értéket kap; az összes többi egy hellyel balra lép; a művelet tkp. 2-vel való szorzást jelent, s →túlcsordulást okozhat, ha a második bitpozícióról értékes bit vesztődik el. Jobbra végzett ~nél a második bitpozíció az előjelbit értékét kapja, az utolsó bit pedig elvész (tkp. 2-vel való osztás történik).

aritmetikai művelet (arithmetical operation) – szűkebb értelemben a négy alapművelet valamelyike, tágabb értelemben viszont minden olyan művelet, amely egy →regiszter v. memóriarekesz tartalmát alkotó bitsorozatot átalakít. Ilyenek pl. a különböző típusú →léptetések, de nem ~ egy rekesz v. regiszter átmásolása egy másik regiszterbe v. rekeszbe. A részt vevő operandusok típusától függően az ~ek lehetnek →fixpontosak v. →lebegőpontosak.

aritmetikai reláció (arithmetical relation) – →logikai kifejezésekben v. döntésekben használható olyan →aritmetikai művelet, amelynek elvégzése után logikai eredmény keletkezik. Tkp. két aritmetikai operandus öszszehasonlításán alapszik, s arra ad választ, hogy az első operandus nagyobb, kisebb, nagyobb v. egyenlő, kisebb v. egyenlő, ill. nem egyenlő a második operandussal. Eredménye mindig az igaz v. hamis igazságérték.

aritmetikai társprocesszor (arithmetical coprocessor) – →személyi számítógépek →alaplapján esetleg megtalálható →mikroprocesszor, amelynek alapvető feladata a →lebegőpontos műveletek elvégzése (tkp. a nagygépek →lebegőpontos aritmetikai egységének megfelelő funkcionális egység).

AROM (Associative Read-Only Memory) – csak olvasható →asszociatív memória (vagyis csak tartalma révén címezhető →permanens memória).

ASCII-kód (ASCII-code) – a számítógépes rendszerekben használatos egyik legelterjedtebb kódolási forma az angol ábécének, a decimális számjegyeknek, írásjeleknek, aritmetikai és logikai műveleti jeleknek, valamint az adatátviteli vezérlés jeleinek ábrázolására. 7, ill. 8 bites változata is ismeretes.

ASCIIZ-füzér (ASCIIZ string) – változó hosszúságú alfanumerikus karakterláncok ábrázolására és kezelésére használt forma: a füzér végét az utolsó tényleges karakter után következő nulla tartalmú →bájt jelzi (→füzér [adat]típus).

aszinkron (asynchronous) – olyan jelenség alapvető tulajdonsága, amelynek időben lefolyó állapotváltozásai teljesen függetlenek a viszonyítás alapjául szolgáló jelenség(ek) állapotváltozásaitól.

aszinkron adatátvitel (asynchronous communication) – a távolsági adatfeldolgozásban elsősorban a kis sebességű, dialógus jellegű üzenetváltást végző →adatállomások átviteli módja. ~ben az információ karakterenként kerül továbbításra: minden karaktert startjel előz meg és stopjel követ, ezért ezt az átviteli módot start-stop átvitelnek is nevezik. Az adó és vevő adatállomások csak a start-stop jelek közötti intervallumban vannak egymáshoz szinkronizálva. A vevőállomás ismeri az adási frekvenciát, és így a startjel hatására elinduló vevőoldal a helyi óra impulzussorozatával mintavételezheti a vett jel minden bitjét. Adásszünetben a stopjelnek megfelelő állapot van a vonalon.

Az ~ előnye, hogy olcsó berendezésekkel is megvalósítható, hátránya viszont, hogy kevésbé használja ki az adatátviteli vonalat.

aszinkron csatorna (asynchronous channel) – olyan →kommunikációs csatorna, amely adatok átvitelére képes, de időzítő információkéra nem.

aszinkron jel (anisochronous signal) – olyan jel, amely semmilyen órajellel nincs kapcsolatban, és amelynél a jelátmenetek bármikor bekövetkezhetnek. Az ~ek használata jellemző a legtöbb klasszikus távíró-telex összeköttetésre.

aszinkron modem (asynchronous modem) – olyan →modem, amely nem képes időzítő információk átvitelére. Nincs szinkronizálva azzal az →adatvégállomással, amelyhez kapcsolódik, így erre hárul minden időzítő feladat.

aszinkron számítógép (asynchronous computer) – olyan →digitális számítógép, amelynek vezérlésére az ún. aszinkron módszert alkalmazzák. Ennek lényege, hogy valamely részművelet indítását egy megelőző részművelet váltja ki annak befejezésekor v. esetleg korábban, amikor a végrehajtandó részművelet elvégzéséhez szükséges →erőforrások már felszabadultak, v. a folyamatban levő részművelet nem veszi azokat igénybe (szemben a →szinkron számítógéppel, amelynél a részműveleteket belső óra vezérli). Ilyen alapon időzítő impulzusokra nincs szükség, s a számítási sebesség is nagyobb. Az aszinkron vezérlési mód azonban olyan versenyhelyzeteket hozhat létre, amilyenek szinkron vezérlésnél nem állhatnak elő. Ezért teljesen ~et ritkán terveznek, annál gyakoribb viszont, hogy egy egészében szinkron működésű számítógép egyes részei aszinkron módon működnek közre.

asszembler (assembler) – olyan →fordítóprogram, amely →gépi kódra fordít le valamilyen →asszembler nyelven megírt programot. Fő funkciói: a szimbolikus jelekkel írt utasításkódok helyettesítése gépi kódokkal, valamint a szimbolikus címek átalakítása olyan módon, hogy ezeknek a fordítás után →relatív, ill. →abszolút címek feleljenek meg.

Az ~eket rendszerint kétmenetesekre tervezik, vagyis kétszer olvassák végig a →forrásprogramot (először eredeti, másodszor megfelelő közbenső formában).

Az első menetben folyik a szimbólumok felismerése és értékük megőrzése (→szimbólumtáblázat felépítése), a gépi kódú utasítások hosszának meghatározása, az →utasí-

tásszámláló megfelelő növelése, az egyes →asszembler direktívák felismerése és végrehajtása, a numerikus és alfanumerikus konstansok összegyűjtése.

A második menetben történik az utasításkódok generálása, a szimbólumok behelyettesítése a megfelelő értékekkel, a még teljesen fel nem dolgozott direktívák végrehajtása.

asszembler direktíva (assembler directive) – →asszembler nyelvekben használatos olyan álutasítás, amely az →asszembler számára jelent előírást (elsősorban magának a →fordítóprogramnak a működését befolyásolja), s nem marad konkrét nyoma a lefordított →tárgyprogramban. Az –k rendszerint memóriazónák lefoglalására, az adatok tulajdonságaira, a fordítás menetére, az →átfedési szerkezetek megvalósítására stb. vonatkoznak.

asszembler lista (assembler listing) – az →asszembler által a →fordítás után készített lista, amelyen a program forrásszintű szimbolikus szövege mellett a fordítás eredményeként kapott →gépi utasítások is szerepelnek. Különleges jelentősége van a programok tesztelési fázisában, ahol bizonyos forrásszintű logikátlanságok a gépi program értelmezése nyomán kiszűrhetők.

asszembler nyelv (assembly language) – olyan szimbolikus →programozási nyelv, amelynek utasításkészlete adott számítógépre jellemző, adatstruktúrái pedig közvetlenül a számítógép memóriarekeszeinek és →regisztereinek felelnek meg. Jellegzetessége, hogy fordításkor minden utasítása egyetlen gépi utasítássá alakul át. Egy-egy konkrét megvalósítása jellegzetes →makroutasításokat, függvényeljárásokat és más álutasításokat is tartalmazhat. Vannak bizonyos előnyei a magas szintű programozási nyelvekkel szemben. A programozó logikailag ugyanúgy alakítja ki programját, mintha az →gépi nyelven készülne, de a szimbolikus kódolásnak köszönhetően sokkal kevesebb fáradsággal. A teljes gépi →utasításkészlet kihasználási lehetősége valóban hatékony programozást biztosít. Némi hátrányt jelent az, hogy az – használata alapos ismereteket követel a konkrét számítógépre vonatkozóan.

asszociatív címzés (associative addressing) – memóriarekeszek lokalizálására szolgáló címzési módszer, azok tartalma alapján. –nél a végrehajtandó utasítás címrészében található bitsorozat alkotja azt a keresési kulcsot, amelyet a megfelelő memóriarekesz kikereséséhez felhasználnak. Az – megvalósítható programozott eszközökkel is, de egyes igen gyors folyamatoknál speciális áramköröket használnak (→asszociatív memória).

asszociatív memória (associative memory) – alapvető tulajdonsága, hogy egy adott rekesz azonosítása a saját tartalma alapján történik, vagyis a →cím szerepét a tárolt információ v. annak egy része veszi át. –nál az adatok tárolási sorrendje tetszőleges, hosszadalmas keresési eljárásokra nincs szükség. Elérési ideje jóval kisebb, mint a címezhető memóriáké, de technikai megvalósításának komplexitása miatt aránylag csak kis kapacitással (4-64 Kbájt) használatos. Kompromisszumos megoldásként az –t hagyományos memóriával összekapcsolva szokás használni. Ilyenkor a keresést az –ban végzik, amely a keresési kulcsokat és a megfelelő adatok címezhető memóriabeli címeit tartalmazza, míg a hagyományos memóriában vannak a tényleges adatok. Az – egyik fontos alkalmazási területe a →virtuális címzés hardvermegvalósítása, vagyis a címeknek →abszolút címekké alakítása. Itt az – rekeszei két mezőre vannak osztva: az első a virtuális cím lapmutatója, ami egyben a keresési kulcs szerepét is betölti, a másik mező pedig a keresett →operatív memóriabeli abszolút lapcímet tartalmazza.

asszociatív regiszter (associative register) – olyan gyors működésű →hardveregység, amely tartalma szerint címezhető, ill. így kérdezhető le.

asztali számológép (desk-top computer) – olyan elektromechanikus v. elektronikus berendezés, amely rendszerint a négy alapműveletet, a százalékszámítás, a gyökvonás, a logaritmusszámítás stb. elvégzésére használható. Billentyűzete numerikus adatok bevitelére, ill. a megfelelő műveletek kiválasztására alkalmas, az eredmények kijelzésére pedig katódsugárcsöves v. folyadékkristályos kijelzőt, esetleg papírszalagot használ. A számítástechnika gyors betörése az irodagépesítés területére olyan –eket eredményezett, amelyek már rövidebb programok előállítására is

alkalmasok, sőt egyesek magas szintű nyelvvel (rendszerint →BASIC) is programozhatók.

áteresztőképesség (throughput) – valamely adatfeldolgozó rendszer teljesítményét jellemző mutató: a rendszer által adott idő alatt végrehajtott (meghatározott típusú) programok számát jelenti.

átfedés (overlay) – nagyméretű programok végrehajtására használt programépítési technika. Alapelve az, hogy ugyanazt az →operatív memóriaterületet rendelik hozzá a program különböző részeihez (→átfedési szerkezet, →szegmentálás). Az egymást átfedő programrészek csak akkor kerülnek az operatív memóriaba, amikor rájuk tényleges hivatkozás történik.

átfedési szerkezet (overlay structure) – a programozói memóriagazdálkodás eszköze. Lényege, hogy a programozó különválasztja programjának azokat a funkcióit, amelyekre nem egy időben van szükség, és azokat különálló programegységekkel (→szegmens) hajtja végre. Ezek a végrehajtás alatt az →operatív memória azonos területére töltődnek be, egyik a másik után. Az ~ megvalósítása a →kapcsolóprogram feladata, amelyet a programozó útmutatásai alapján állít elő. Jelentősége elsősorban a fix hosszúságú →partíciókkal dolgozó →operációs rendszereknél van.

áthelyezés (relocation) – mindazoknak a műveleteknek az összessége, amelyek lehetővé teszik, hogy a programozásban használt logikai címeket közvetlenül a végrehajtás alatt alakítsák érvényes →abszolút címekké, biztosítva ezáltal az →operatív memória gazdaságos kiosztását. A programozó a programmodulok kódolásakor →relatív címeket használ.

Statikus ~nél ezeket a relatív címeket a →kapcsolóprogram alakítja abszolút címekké. Ennek a módszernek nagy hátránya, hogy a címek már a betöltés előtt véglegesek, s így a programok csak az előre befagyasztott címekre tölthetők az operatív memóriába.

Dinamikus ~nél a logikai-fizikai címátalakítás csak a program végrehajtásakor valósul meg, s ez lehetővé teszi a memória dinamikus, ismételt és gyors újrakioszthatóságát. A dinamikus ~ egyik megoldási módszere a →bázisregiszterek használata. Itt az abszolút címet mindig a bázisregiszter tartalmának és a címrészekben található →eltolásoknak az összege adja. A módszer hátránya, hogy az egész programot egyetlen összefüggő memóriazónába kell betölteni.

A dinamikus ~ modernebb változata a →lapkezelés, amelynél a programokat megegyező méretű lapokra (2-4 Kbájt) osztják, s ezeket szétszórtan, a pillanatnyilag szabad memóriazónákba töltik. A logikai címeket táblázatok segítségével alakítják abszolút címekké. Mivel nem szükséges, hogy a futó program minden lapja egyszerre az operatív memóriában tartózkodjék, a logikai →címtartomány jóval meghaladhatja a fizikait.

áthelyezhető betöltőprogram (relocatable loader) – →áthelyezhető bináris formájú programok betöltésére szolgáló →betöltőprogram. Legfontosabb feladatai közé tartozik a →címek →áthelyezése, vagyis az összes helytől függő érték végleges kiszámítása, az esetleges →címkonstansok meghatározása, a program betöltése.

áthelyezhető bináris forma (relocatable binary form) – olyan binárisan kódolt forma, amellyel a programok bárhol elhelyezhetők az →operatív memóriában. Általában a különböző nyelvek →fordítóprogramjainak közös →tárgynyelve. Alapvetően kétféle információt tartalmaz: a tárgyprogramhoz tartozó tulajdonképpeni bináris szöveget, valamint a szimbolikus hivatkozásokra és az →áthelyezés megvalósítására vonatkozó információkat.

Az áthelyezés (→relatív címek átalakítása →abszolút címekké) különböző szinteken valósulhat meg; közreműködik benne a →kapcsolóprogram, a →betöltőprogram, sőt a végrehajtás idejére is kitolható (→bázisregiszterek használatával, ill. a →virtuális címzés megvalósításával).

áthelyezhető kifejezés (translatable expression) – →asszembler nyelven kódolt programokban valamely operandus →címének meghatározására használatos olyan aritmetikai kifejezés, amelynek értéke a program →áthelyezésénél módosítást szenved (szemben az →abszolút kifejezéssel). Az ~ben az áthelyezhető elemek csak összeadásban és kivonásban szerepelhetnek, számban páratlanok s – egy kivételével – páronként ellenkező előjellel csoportosíthatók kell hogy legyenek.

áthelyezhető program (relocatable program) – →áthelyezhető bináris formában kódolt, rendszerint nem közvetlenül végrehajtható program. Alapvető jellegzetessége, hogy egyetlen hivatkozása sem vonatkozik →abszolút címre. A →fordítóprogramok által előállított →tárgyprogramok rendszerint ~ok.

átlagos elérési idő (average access time) – →soros hozzáférésű memóriáknál a (véletlenszerűen választott című) rekesz átlagos hozzáférési ideje s egyben a hozzáférési idő várható értéke.

atom (atom) – valamely adott típusú feldolgozás szempontjából oszthatatlan információmennyiség (adat). Az elnevezés a →LISP programozási nyelvből származik, ugyanis ott az ~ok olyan oszthatatlan objektumokat képviselnek, amelyekhez attribútumokat lehet rendelni, s amelyek között bizonyos összefüggések állapíthatók meg. Más megközelítésben sokszor a strukturálatlan adatokat is ~oknak nevezik.

attribútum (attribute) –1. Valamely →szintaktikai kategóriát kísérő, jellemző információ, amely kiemelkedő jelentőségű a programok →fordításánál. Az ~ lehet explicit →deklaráció tárgya, de következhet szövegkörnyezetből is. A →szintaktikai analízis egyik legfontosabb feladata a programokban szereplő objektumokhoz tartozó összes ~ összegyűjtése, rendszerezése és ellenőrzése (→attribútumnyelvtan). – 2. Állománykezelésnél: jelzőbit (pl. „csak olvasható" állomány).

attribútumnyelvtan (attribute grammar) – →programozási nyelvek →szintaxisának és →szemantikájának leírására D. Knuth által javasolt formális rendszer. Knuth szerint a szemantikát közvetlenül a szintaxishoz kapcsolt →attribútumok segítségével kell megadni. A →szintaxisfa minden csomópontja →metaváltozót, →nemterminális szimbólumot jelképez, s ezekhez kötődnek az attribútumok. Ugyancsak Knuth meghatározása szerint az attribútumok két csoportba oszthatók. Az öröklött attribútumokat az illető szintaktikai egység felülről, vagyis a gyökérelem felől kapja, a szintetizált attribútumokat pedig alulról, a →terminális szimbólumok felől kapott attribútumokból és az öröklött attribútumokból számítja ki, valamilyen függvényeljárás segítségével. Az ~ alapvető jelentősége az, hogy szemantikai leírást kapcsol a szintaktikai leíráshoz, s ezáltal a →fordítóprogramok előállításánál a szemantika kezelésére pontos szabályokat biztosít.

átvitel (carry) – összeadásnál keletkezik valamely helyértéken akkor, ha az összeadandó két számjegy összege nem kisebb, mint a használt számrendszer alapszáma. Az ~t, amely – az alkalmazott számrendszertől függetlenül – két szám összeadásánál mindig 1, a következő helyértéken az összeghez kell adni.

átviteli rétegprotokoll (transport layer protocol) – →hétréteges referenciamodell

átviteli utasítás (transfer instruction) – olyan művelet, amelynek során valamely memóriarekesz v. →regiszter tartalma másik rekeszbe v. regiszterbe másolódik át.

autokód (autocode) – a formális →gépi nyelvek egyszerűbb változata, a tulajdonképpeni gépi nyelv és az →asszembler nyelvek között átmenetet képező programozási nyelv. Egyes utasításai még közel állnak a gépi kódhoz, de a ciklusutasítások, feltételes utasítások már egy asszembler nyelv megfelelő utasításaihoz hasonlítanak. Az ~ban írott programokat speciális →fordítóprogram alakítja át gépi nyelvre.

automata (automaton) – információfeldolgozásra kifejlesztett berendezések és eljárások tanulmányozására szolgáló absztrakt matematikai modell. Formális definíciója szerint olyan (A,S,V,k,f,g) hatos rendszer, ahol A a bemenő szimbólumok halmaza, S az ~ lehetséges állapotainak halmaza, V a válaszként adható szimbólumok halmaza, k az ~ kezdő állapota [$k \in S$], g az a leképzés, amelylyel az ~ az S halmazt a V-re leképezi [$v=g(s)$, $v \in V$, $s \in S$], f az a leképzés, amellyel az ~ az A · S Descartes-féle szorzatot az S halmazra képezi le [$g=f(s,a)$, $a \in A$, $s \in S$, $g \in S$].

Ha az A,S és V véges elemszámú halmazok, az ~ véges. Ha az f leképzés jól meghatározott, az A és S halmazokhoz tartozó minden elempárhoz egyetlen S-hez tartozó elemet határoz meg, az ~ determinisztikus.

automatikusan javított hiba (automatically cleared failure) – valamely számítógéprendszerben előállott hiba, amelyet a rendszer felismer és azonnal kijavít anélkül, hogy

róla a felhasználó tudomást szerezne. Ilyen hibák lehetnek pl. az adatátvitelben fellépő →paritáshibák (megfelelő hibajavító kódolásnál), a →periferikus művek egyes hibái stb. Hardverelemeknél az ~ nem tulajdonképpeni javítás, hanem a hibásnak bizonyult egység helyettesítése más hasonló típussal. →Mágneslemezeknél pl. a rendszer újabb eredménytelen próbálkozások után esetleg kiiktatja a hibás →sávot, és tartalék sávval helyettesíti azt (természetesen csak kiviteli műveleteknél).

automatikus ismétléskérés (automatic repeat request) – →adatátviteli hálózatokban az egyik legjobban elterjedt hibakorlátozó stratégia. Ennél az eljárásnál általában hibajelző kódolót és dekódolót használnak. A dekódoló által jelzett hibákról a vevőoldal automatikusan hibajelentést küld vissza az adónak, mire az megismétli a hibásnak vélt üzenetet. Az ismétlések száma lényegesen csökkenthető, ha →hibajavító kódot használnak, és csak a nem javítható hibák észlelésénél kér ismétlést a vevőállomás. Az ~sel dolgozó rendszerek többségénél az adóállomás a helyes vételről →visszaigazolást igényel, csak kevés rendszer tekinti a →negatív visszaigazolás hiányát hibátlan átvitelnek.

automatikus kötetfelismerés (automatic volume recognition) – →kötegelt feldolgozással működő →operációs rendszereknél az a lehetőség, hogy a gépkezelő a megfelelő →periferikus művekre (→mágneslemez- v. →mágnesszalag-meghajtó) előbb készíthet fel →szabványos címkékkel ellátott →köteteket, mint ahogy azokra a felhasználói programoknak tényleges szüksége lenne. A kötetet tartalmazó meghajtó bekapcsolásakor →megszakítást hoz létre, amit az operációs rendszer úgy dolgoz fel, hogy az illető kötet azonosítójáról és a meghajtó periferikus mű fizikai címéről feljegyzést készít. Így majd automatikusan felismeri azokat a már rendelkezésre álló köteteket, amelyekre az éppen induló munkalépésnek szüksége van. Ezzel a módszerrel megtakarítható az az idő, amit a rendszer egyébként várakozással töltene, amíg a gépkezelő – csak a tényleges igény után – megkeresi és felkészíti a kért kötetet.

automatikus változó (automatic variable) – olyan →változó, amely csak az őt tartalmazó →blokk aktivizálásakor kap tényleges tárolóhelyet, s a blokk végrehajtása után ez a terület felszabadul. Ez azt jelenti, hogy a blokk következő aktivizálásakor az ~ az →operatív memória másik területére kerülhet, s ezért annak előző értékére nem lehet számítani (→dinamikus változó). A →blokkstruktúrájú programozási nyelvek (pl. →PL/1, →Pascal stb.) jellegzetes kategóriája.

axiomatikus szemantika (axiomatic semantics) – →programozási nyelvek →szemantikájának axiómák (bizonyítás nélkül igaznak tekintett kijelentések) segítségével történő leírása. Így pl. egy →utasítás jelentését olyan axióma határozza meg, amely megadja az illető utasítás utáni állapotot a végrehajtás előtti állapot függvényében. Fontos módszer a programok helyességének bizonyításánál.

azonos idejű üzemmód (real-time operation) – →valós idejű üzemmód

azonosító (identifier) – →programozási nyelvek →szintaktikai kategóriája: →konstansok, →változók, →eljárások és →függvények nevét képező jelsorozat. Az ~ teszi lehetővé, hogy a →program különböző objektumaira hivatkozni lehessen. Szinte valamennyi programozási nyelvben előfordulnak standard ~k is, amelyeknek előre definiált jelentésük van (rendszerint általánosan használt függvényekre, eljárásokra stb. vonatkoznak). A programozási nyelvek →lefoglalt szói általában nem használhatók ~ként.

azonosító főrész (identification division) – →COBOL programoknak az a része, amely az illető program nevét, szerzőjét, keltezését, fordításának dátumát, esetleg a program rövid leírását, az →erőforrásokra vonatkozó megjegyzéseket stb. tartalmazza.

B

B - 1. →Hexadecimális ábrázolás jelkészletének eleme, amely a decimális 11 értéknek felel meg (bináris megfelelője: 1011). – 2. Magas szintű →programozási nyelv, amelyet Ken Thomson fejlesztett ki az →UNIX →operációs rendszer első megvalósításában. A →BCPL programozási nyelv továbbfejlesztésének tekinthető, s ugyanakkor a később megvalósított →C nyelv előfutárának. Elsősorban →fordítóprogramok és más rendszerprogramok írására használták, de a →strukturált programozás elveit megvalósító eszközeivel programfejlesztésre is alkalmasnak bizonyult.

Backus-Naur Forma (Backus-Naur Form) – J. W. Backus által 1959-ben javasolt →metanyelv, amelyet 1963-ban használtak először →programozási nyelv (→ALGOL) →szintaxisának formális leírására. A definiálandó →nyelvtan →terminális szimbólumait metakonstansoknak, →nemterminális szimbólumait metaváltozóknak nevezik, a metaszabályok leírására pedig a következő metaszimbólumokat használják: <, >, ::=, |. A <és> metaszimbólumok (metazárójelek) a metaváltozókat zárják közre; a ::= metaszimbólum jelentése az, hogy a bal oldalon álló metaváltozó egy szöveg származtatásakor helyettesíthető a jobb oldalon álló metakifejezés bármelyik alternatívájával; a | metaszimbólum a lehetséges alternatívákat választja el egymástól. Így pl. a „decimális számjegy" →szintaktikai kategóriát a következő metaszabály írja le:

<decimális számjegy>::=
::=0|1|2|3|4|5|6|7|8|9

értelmezése pedig a következő: decimális számjegy lehet a jobb oldalon szereplő tíz alternatíva bármelyike.

Az →azonosítót az alábbi metaszabály írja le a programozási nyelvek nagy többségénél:

<azonosító>::= <betű> |
| <azonosító> <betű> |
| <azonosító> <számjegy>

vagyis azonosító lehet egy betű v. bármilyen (betűből és számjegyből álló) karaktersorozat, amelynek első karaktere betű.

A ~ igen jelentős a gyakorlatban, elsősorban a →szintaktikai analízis szempontjából, mert könnyen kezelhető, automatizálható eszközt jelent a →fordítóprogramok megvalósításában.

bájt (byte) – rendszerint 8 →bitből (plusz egy hibaellenőrző bitből) álló bináris vektor, amely az információfeldolgozás jelenleg leggyakrabban használt önálló egysége. Egy ~tal ábrázolható egy →alfanumerikus karakter, v. két →hexadecimális számjegy, v. két →tömörített ábrázolású decimális számjegy, v. egy →zónázott ábrázolású decimális számjegy. A 8 információhordozó bit révén a ~ lehetséges kódválasztéka 256 különböző jel.

bájtbeültetés (byte stuffing) – →adatátvitelben használatos eljárás. Az előfizetők a hálózathoz olyan egységeken keresztül kapcsolódnak, amelyek négy különböző jelsebességgel dolgozhatnak: 0.6, 2.4, 9.6 és 48 Kbit/sec. Ezeknél az értékeknél kisebb sebességet is lehet használni néhány „üres" →bájtnak a bájtáramba való behelyezésével, ami által a közepes adatátviteli sebesség kisebb lesz, mint a →kommunikációs csatorna sebessége. A ~ lehetővé teszi a hálózat szinkronizmusának megőrzését annak ellenére, hogy bármilyen sebességű adatcsatorna előfordulhat.

bájtmultiplexelés (byte multiplexing) – az →időmultiplexelés olyan formája, amelynél az egyes alcsatornákból egy-egy →bájt vesz részt az átvitelben, és a különböző alcsatornák bájtjai egymás utáni időegységekben követik egymást.

bájtparallel átvitel (byte-parallel transmission) – adatátvitelben használható igen gyors, de költséges módszer, amelynél egy nagyobb egység (pl. →szó) →bájtjait párhuzamosan viszik át, akárcsak a bájtokat alkotó →biteket. Általában a →számítógépek funkcionális egységeit összekötő →sínrendszereknél használják.

bájtsoros átvitel (byte-serial transmission) – a →bitpárhuzamos átvitellel lényegében azonos átviteli módszer, amelynél egyegy →bájt →bitjeit párhuzamosan viszik át 9 elemi csatornán (8 információbit + 1 ellenőrzőbit), a bájtokat viszont sorosan, egyiket a másik után.

bájtszervezésű számítógép (byte-organized computer) – olyan →digitális számítógép, amelynek alapvető információegysége a →bájt. Utasításai elsősorban bájtokra hivatkoznak, de vonatkozhatnak olyan információs egységekre is, amelyeknek hossza a bájt egész számú többszöröse (pl. félszó, →szó stb.).

balra igazodó ábrázolás (left justified representation) – →alfanumerikus adatok olyan tárolási módja, amelyben az adat felépítő karakterei a számukra fenntartott zónát annak bal első bájtjától kezdve töltik ki. Ha az adat karaktereinek száma kisebb, mint a kiosztott zóna bájtjaié, a ki nem használt bájtokat szóközök töltik ki a zóna jobb széléig (szemben a →jobbra igazodó ábrázolással). A magas szintű programozási nyelvekben speciális előírást használnak a ~ jelzésére (pl. a →COBOL nyelvben ilyen a JUSTIFIED LEFT).

„balról jobbra" szabály („from left to right" rule) – aritmetikai műveletek végrehajtási sorrendjét meghatározó, a →precedenciaszabályt kiegészítő szabály, amely szerint: ha azonos elsőbbségű műveleti jel van az operandus jobb és bal oldalán, akkor az operandus a tőle balra álló műveleti jelhez fog tartozni. Más megközelítésben: az azonos precedenciaosztályba tartozó műveleteket balról jobbra kell kiértékelni. A ~ szerint az A/B·C kiértékelési sorrendje tehát: (A/B)·C. Sok →fordítóprogram nem követi ezt a szabályt, és ez könnyen →mellékhatásokhoz vezethet, amelyek viszont bármikor elkerülhetők a megfelelő zárójelezéssel.

BASIC (Beginner's All-purpose Symbolic Instruction Code) – könnyen megtanulható, főleg a számítástechnikával először kapcsolatba kerülők számára kifejlesztett magas szintű →programozási nyelv, elsősorban →párbeszédes üzemmódra. Különleges jelentőségre tett szert a →mikroprocesszorok, ill. a →személyi számítógépek megjelenésével és elterjedésével. A legtöbb mikroszámítógépre implementálták, így több tucat dialektusa létezik. Egy ~ program sorszámmal ellátott utasítások sorozatából áll, amelyeket interaktív módon lehet javítani, lefordítani v. esetleg rögtön végre is hajtani. A sorszámokkal lehet hivatkozni a logikai és a vezérlőutasítások megfelelő soraira, ami azt is jelenti, hogy a sorok bármilyen sorrendben írhatók, a végrehajtás sorrendjétől függetlenül. Gyakorlati megvalósításaiban nemcsak nyelvről, hanem inkább ~ rendszerről szokás beszélni, ebbe a tágabb fogalomba beleértve a programok írásának, javításának, tesztelésének és végrehajtásának eszközeit is (modernebb kivitelezésben már: →programozási környezet).

baud (baud) – az →adatátviteli sebesség egyik egysége, amely az adatvonalakra vonatkoztatva az egy másodperc alatt átvihető elemi vonalállapotok számát határozza meg.

báziscím (base address) – a →dinamikus áthelyezés, ill. a →relatív címzés megvalósítását teszi lehetővé. Olyan programegység →betöltési címe, amelyben az →abszolút címet a végrehajtás pillanatában →címszámítás útján állítják elő. Értékét a →bázisregiszterben található ~ és az utasítás →címrészében található →eltolás összege adja. Jelentősége van a nagy kiterjedésű →operatív memóriák címzésében is, ugyanis az eltolást tartalmazó címrész az esetek többségében csak 2 →bájt hosszúságú, ami viszont csak 64 Kbájt címzését teszi lehetővé. A ~ és az eltolás összegét viszont hosszabb bitvektoron is lehet ábrázolni (gyakran 24 bit v. ennél is több), ami már 16 Mbájt címzésére is elegendő.

bázisnyelv (base language) – a →kiterjeszthető nyelvekkel kapcsolatos fogalom. Azokat a minimális nyelvi eszközöket foglalja magában, amelyek minden feladat megoldásához feltétlenül szükségesek, s amelyek lehetőséget biztosítanak további nyelvi eszközök létrehozására is.

bázisregiszter (base register) – olyan →regiszter, amely a →báziscím tárolására alkalmas, vagyis tartalmazza annak a memóriatartománynak a kezdőcímét, ahova a programegységet végrehajtás céljából betöltötték. A →dinamikus áthelyezés egyik megvalósítási eszköze, ugyanis a programegység a ~ tartalmának megváltoztatásával egyszerűen, más változtatás nélkül áthelyezhető az →operatív memória egyik tartományából a másikba.

BCPL (Basic Combined Programming Language) – magas szintű →programozási nyelv, amelyet elsősorban →fordítóprogramok írására fejlesztettek ki úgy, hogy egyesítse a magas szintű és az →asszembler nyelvek előnyeit. A ~-program →absztrakt számí-

tógépen működik, definícióját standard fordítóprogram adja meg (ezt nagyon egyszerű nyelven – INTCODE – programozták, s ezáltal a ~ fordítóprogramja kis befektetéssel bármely konkrét számítógépen megvalósítható).

A ~ →[adat]típus nélküli nyelv, egyetlen összetett adatszerkezete a szósorozat. Egyik legérdekesebb vonása az a szemlélet, ahogyan az adatokat megközelíti: az absztrakt gép memóriáját szósorozatnak tekinti. Minden szónak két attribútuma van: a jobbértéknek nevezett tartalom, és a balértéknek nevezett cím. Speciális operátorokkal lehet hivatkozni egy-egy változó v. kifejezés bal-, ill. jobbértékére. Az elemi műveletek operandusa és eredménye a gépi szó.

Más jellemző vonásai a →blokkstruktúra, a →strukturált programozást elősegítő utasításkészlet, a →rekurzivitás programozhatósága stb. Sok alapötlete szolgált a későbbiekben igen népszerűvé vált →C programozási nyelv megvalósítására.

beágyazott kód (embedded code) – →beágyazott rendszerek vezérlésére előállított program. Rendszerint idegen fejlesztési környezetben állítják elő speciális →programozási nyelvek segítségével (pl. →Ada, →CHILL stb.). A fejlesztési rendszernek lehetőséget kell biztosítania a célberendezés megfelelő leírására, hogy az előállított ~ megfelelően működhessen a beágyazott rendszer speciális →hardverjén. A ~ban nemcsak a számítógépen belül lezajló eseményeket kell megfelelő módon kezelni, hanem a számítógép és az általa vezérelt környezet közötti kölcsönhatásokat (→kivételeket) is.

beágyazott rendszer (embedded system) – olyan rendszer, amelyben a számítógép közvetlenül kapcsolódik ahhoz a berendezéshez, amelyre felügyel, v. amelyet irányít. Ilyen rendszerek pl. a számjegyvezérlésű szerszámgépek, a navigációs és irányítási rendszerek, a tárolt programú távbeszélő rendszerek, a termelés- és folyamatirányítási rendszerek stb. Programozástechnikai szempontból különleges szerepük van, ugyanis megbízhatóan kell kezelniük előre nem látható eseményeket (→kivételeket), időzítési és ütemezési feladatokat, versenyhelyzeteket kell megbízhatóan megoldaniuk stb. A ~ alapvető jellemzői a hosszú távú karbantarthatóság, a módosíthatóság, kiterjeszthetőség stb.

bécsi definíciós nyelv (Vienna definition language) – →programozási nyelvek formális leírására az 1960-as években az IBM bécsi laboratóriumaiban kifejlesztett módszer. Első gyakorlati felhasználása a →PL/1 leírása volt. A ~ az →absztrakt számítógép fogalmán alapszik: a leírandó nyelv részére olyan absztrakt gépet definiálnak, amely értelmezőként működik. A ~ a →fával ábrázolható adatszerkezetekhez speciális szelektoreljárásokat és konstrukciós műveleteket rendel.

becslés (estimation) - valamely mennyiség közelítő meghatározása számítással v. empirikus eljárással. A numerikus analízis, valamint a számítógépes feladatmegoldás egyik fontos feladata a →hibabecslés.

beégetés (burning) – az a művelet, amellyel →programozható permanens memóriába programot rögzítenek (elektromos impulzus, ultraibolya besugárzás stb. segítségével).

beépített makroutasítás (built-in macroinstruction) – →rendszer-makroutasítás

befogadó nyelv (host language) – olyan →eljárásorientált magas szintű →programozási nyelv, amelybe az →adatleíró, ill. az →adatkezelő nyelv utasításait beépítik. Legtöbbször a →COBOL, →PL/1 v. →FORTRAN nyelvek →fordítóprogramjait bővítik ki az új típusú utasítások értelmezési képességével. Másik lehetőség esetleg az →előfordítás, amely az eredeti forrásprogramot a szabványos fordítóprogram által elfogadható formára alakítja át.

bejárási stratégia (routing strategy) – →fa szerkezetű adatstruktúrák bejárására használatos módszer. A →bináris fa bejárásának három módozata ismeretes.

1) A preorder stratégia szerint vizsgálják a fa gyökérelemét, bejárják a balra leágazó részfát, majd a jobbra leágazó részfát.

2) A postorder stratégia szerint először bejárják a balra leágazó részfát, utána a jobbra leágazó részfát, s csak ezután kerül sor a gyökérelem megvizsgálására.

3) Az enorder stratégia szerint először a balra leágazó részfát járják be, majd megvizsgálják a gyökérelemet, s utána járják be a jobbra leágazó részfát.

bejelentkezés (log in) – →párbeszédes üzemmódban dolgozó rendszereknél azoknak a szabályoknak, műveleteknek az összessége, amelyeknek segítségével az →operációs rendszernek bizonyítják, hogy jogos felhasználó kéri a számítógép szolgáltatásait. Sikeres ~ után a felhasználó rendelkezésére állnak a rendszer azon →erőforrásai, amelyeknek használatára jogosítványa van.

belépési pont (entry point) – valamely hívható programegységnek az a pontja, ahonnan aktivizáláskor annak végrehajtása megkezdődik. Meghatározására az adott →programozási nyelv ad pontos előírásokat. Rendszerint →külső hivatkozásban szereplő →globális szimbólum, amelyet a →kapcsolóprogram old fel. Implicit definíciója általában a programegység első végrehajtható utasítására vonatkozik.

belső adat (internal data) – olyan adat, amely a feldolgozás pillanatában a programmal együtt az →operatív memóriában található. Lehet kezdeti érték (a programozó határozta meg a program kódolásakor), létrejöhet egy előbbi művelet eredményeként, v. beviteli utasítás útján kerülhet be külső →adathordozóról. Alapvető jellegzetessége, hogy közvetlenül címezhető, tehát a →központi vezérlőegység által feldolgozható.

belső kód (internal code) – →gépi nyelv

belső memória (internal memory) – →operatív memória

belső rendezés (internal sorting) – olyan →rendezés, amely teljes egészében végrehajtható az →operatív memóriában. Kihasználja azt a lehetőséget, hogy az operatív memóriában az adatok közvetlenül címezhetők. Legismertebb módszerei a →beszúrásos rendezés, a →helycserés rendezés (buborékrendezés), a →kieséses rendezés, a →kiválasztásos rendezés, a →kvadratikus rendezés, az →összeválogatásos rendezés, a →szétszórásos rendezés, →gyorsrendezés, →Shell-rendezés.

bemeneti [adat]állomány (input file) – az az →[adat]állomány, amely a feldolgozásra szánt adatokat tartalmazza. Kiemelt jelentőségű a →kötegelt feldolgozásnál a standard ~, amely a feldolgozásra váró →munkákat és a feldolgozást irányító →vezérkártyákat tartalmazza. A standard ~ rendszerint →szekvenciális [adat]állomány.

bemeneti egység (input unit) – olyan →periferikus mű, amely alkalmas adatoknak az →operatív memóriába való bevitelére. Lehet önálló egység (pl. →lyukkártyaolvasó, lyukszalagolvasó) v. alrendszere egy →b/k egységnek (pl. →mágnesszalag, →mágneslemez). Kiemelt jelentőségű a →kötegelt feldolgozásnál a standard ~, amely a →háttérpartíció dedikált egysége, s rendszerint logikai szinten, standard →bemeneti [adat]állományként kezelik.

bemenő ábécé (input alphabet) – valamely →automata bemenetén jelentkező szimbólumok halmaza v. a →fordítóprogram által lefordítandó →forrásprogram szimbólumainak összessége.

beolvasási hiba (reading error) – külső →adathordozókon tárolt adatok beolvasása közben fellépő hibatípus, amely a →periferikus mű, a →multiplexcsatorna, ill. a →szelektorcsatorna hibás működésének következménye. Kiszűrése elsősorban az →operációs rendszer feladata, amely a →központi vezérlőegységen keresztül állandó felügyelet alatt tartja a beviteli művelet végrehajtását. A ~ rendszerint az adatot igénylő program végrehajtásának rendellenes befejezéséhez vezet. Mágneses adathordozóknál, ha a beolvasási művelet megismételhető, csak többszöri próbálkozás után jelez ~t az operációs rendszer.

bérelt vonal (leased line) – nyilvános hálózat olyan →kommunikációs vonala, amely bérlet formájában egyetlen felhasználó kizárólagos rendelkezésére áll.

berendezéskiértékelés (benchmark) – számítástechnikai rendszerek működési paramétereinek meghatározására használt módszer. Rendszerint erre a speciális célra előállított programcsomag, amely a számítógépnek egy adott területen történő felhasználásában nyújtott minőségi mutatóit méri. A programok százalékos arányban tartalmaznak az illető felhasználási területre jellemző műveleteket.

besorolási kulcs (order key) – olyan →kulcs, amely meghatározza adatelemek elhelyezési sorrendjét valamely rendezett →heterogén adatstruktúrában (→[adat]állományban, →táblázatban stb.).

beszúrásos rendezés (insertion sorting) – a →rendezésnek az →operatív memóriában

megvalósítható módszere. Lényege, hogy a rendezendő halmaz elemeit egyenként vizsgálják és viszik át a rendezett halmaznak fenntartott területre. Az éppen vizsgált elemet úgy illesztik a sorrendjének megfelelő helyre a rendezett halmazba, hogy az utána következő elemeket eggyel eltolják. Gyorsítható az eljárás azzal, hogy az első elhelyezendő elemet nem a rendezett, de még üres halmaz elejére teszik, hanem annak közepére, és a többi beillesztendő elem helyét →bináris kereséssel határozzák meg.

betöltés (loading) – az az összetett művelet, amelynek révén valamely →program végrehajtás céljából az →operatív memóriába kerül. Legfontosabb funkciói: a →fordítóprogramok v. a →kapcsolóprogramok által valamely háttértárban rögzített →tárgyprogramok átalakítása közvetlenül végrehajtható formába, szabad memóriaterület kijelölése a programok számára, a →bázisregiszterek feltöltése, a programok tényleges bemásolása az operatív memóriába.

Az →operációs rendszerek speciális rendszerprogramokat tartalmaznak a ~ végrehajtására (→betöltőprogram).

betöltési cím (loading address) – az →operatív memóriának az az →abszolút címe, ahova egy programot (programegységet) végrehajtás céljából betöltenek. A ~et rendszerint egy →bázisregiszter tartalmazza.

betöltési idő (loading time) – a →betöltőprogram egyszeri végrehajtásának időigénye, amelynek során egy adott programot a →háttértárból az →operatív memóriába betölt.

betöltőprogram (loading program) – az →operációs rendszerekhez tartozó olyan rendszerprogram, amely a →betöltés műveletét elvégzi. Alapvető feladata a →tárgyprogramok betöltése és a végrehajtás elindítása, de járulékos feladatként az →áthelyezés bizonyos formáit is elvégezheti (→áthelyezhető betöltőprogram).

betöltőszektor (boot sector) – az →operációs rendszerek →betöltését elindító →gépi kódolású program helye a →mágneslemezen (→elsődleges betöltés). Az IBM-szabvány szerint szervezett →hajlékonylemezeknél a 0. oldal 0. →sávjának 1. →szektora, még ha a lemezen nincs is operációs rendszer.

betűkészlet (font) - jellemzőikben (nagyság, vastagság, grafikai jelleg stb.) összetartozó →karakterek készlete. A ~et rendszerint →bitleképzéses módban tárolják az →operatív memória erre a célra fenntartott zónájában, ahonnan ki lehet vetíteni a képernyőre v. ki lehet nyomtatni.

betű-szám váltás (letters-figures shift) – →billentyűzettel ellátott számítástechnikai berendezéseknél (írógépek, telexgépek, →terminálok stb.) a kívánt karakterkészlet kiválasztása. A billentyűzeten ugyanis minden nyomógombon általában két jel van: az egyik rendszerint betű, a másik szám v. különleges karakter. A ~ra külön billentyű áll rendelkezésre (SHIFT). A modern távírógépeknek nincs átkapcsoló billentyűje, a szükséges átkapcsolójelet automatikusan képezi és a többi jelhez hozzáteszi (Baudot-kód). Az átkapcsolójel más jelekhez hasonlóan vihető át, és a fogadó távgépíróban mindig a következő jelcsoportra vonatkozó állást hoz létre.

billentyűzet (keyboard) – számítógéppel folytatott párbeszédre v. egyedi adatbevitelre szolgáló →periferikus mű (→terminál) funkcionális egysége. Hasonló a hagyományos írógépek billentyűzetéhez, csak rendszerint nagyobb jelkészlettel rendelkezik. A billentyűk három kategóriába sorolhatók. A karakterbillentyűk lenyomása egy-egy alfanumerikus, ill. speciális karakter megfelelő bináris kódját juttatja a számítógépbe. A funkcionális billentyűk lenyomása egy-egy művelet elindítását váltja ki. A vegyes használatú billentyűk önállóan karakterbillentyűk, valamely funkcionális billentyűvel (Ctrl, Alt) együtt lenyomva további műveleteket indítanak el.

bináris ábrázolás (binary representation) – számoknak a 2 hatványai szerint rendezett felírása, ami az

$$a_n \cdot 2^n + a_{n-1} \cdot 2^{(n-1)} + \ldots + a_1 \cdot 2^1 + a_0 \cdot 2^0$$

felbontásnak felel meg. Az a_n, a_{n-1}, a_{n-2}, ..., a_1, a_0 együtthatók jelkészlete: $\{0,1\}$. Minthogy a számítógépekben az információ tárolására kétállapotú elemeket használnak, a két jelből álló jelkészlet megfelel ennek a tárolási módnak.

binárisan kódolt decimális kód (binary coded decimal code) – a decimális számrendszer jelkészletének olyan →kódja, amely-

nél a 4 →bitből álló tetrádban a különböző jegyek értékének megfelelő bináris kombinációt feleltetik meg. Ebben a kódolási formában az 1010, 1011, 1100, 1101, 1110, 1111 bináris jelkombinációk értelmetlenek (→áltetrádok).

bináris betöltőprogram (binary loader) – →abszolút bináris formájú programok betöltésére szolgáló →betöltőprogram. Egyetlen feladata a program betöltése az előírt →abszolút címre.

bináris-decimális konverzió (binary-to-decimal conversion) – kettes számrendszerben ábrázolt számok átírása tízes számrendszerbe: a kettes számrendszerben írt szám jegyeivel (0,1) felírják az adott jegyekhez tartozó helyértékeknek megfelelő kettes hatványösszeget.

bináris fa (binary tree) – olyan →fa, amelyben minden csomópontból legfeljebb két él ágazik el, vagyis minden csomóponthoz legfeljebb két részfa csatlakozhat. Egyik fontos alkalmazási területe a →fordítóprogramokban az aritmetikai kifejezések ábrázolása és feldolgozása. Itt minden műveleti jelhez két operandus tartozik. A műveleti jeleket a ~ csomópontjaiban helyezik el, a hozzá tartozó bal és jobb oldali operandust pedig a bal, ill. a jobb oldali részfa határozza meg. Ha ezek a részfák a fa terminálisai, akkor az operandusok a terminálishoz rendelt értékek.

bináris-hexadecimális konverzió (binary-to-hexadecimal conversion) – kettes számrendszerben ábrázolt számok átalakítása →hexadecimális ábrázolásba. Mivel $16=2^4$, minden kettes számrendszerben felírt szám egyben hexadecimális számrendszerben felírtnak is tekinthető, ugyanis minden 4 bit egy hexadecimális jegyet képvisel.

Bináris számot tehát úgy írnak át hexadecimális számmá, hogy a bináris ponttól jobbra és balra a biteket négyes csoportokba fogják össze, és minden ilyen csoportot a megfelelő hexadecimális jeggyel helyettesítenek.

bináris keresés (binary search) – olyan →keresési algoritmus, amely a keresési zóna módszeres felosztásán alapszik. Alapfeltétele, hogy az adatelemek a keresési kulcs szerint fizikailag is szekvenciálisan legyenek rendezve. Ha minden elemet ugyanakkora valószínűséggel kereshetnek, akkor kimutatható, hogy az összehasonlítások átlagos száma minimális, ha a részzónákat minden esetben két egyenlő részre osztják és az összehasonlításokat a zóna középső elemével kezdik. N elemet feltételezve a keresési lépések átlagos száma $S_N = \log_2(N)$.

bináris keresési fa (binary search tree) – →rendezési v. →keresési célokra létrehozott olyan →bináris fa, amelynek i-edik csomópontjához az a_i kulccsal jellemzett információt rendelik. Érvényesek a következő állítások: bármely részfa gyökérelemétől balra leágazó részfában a gyökérelemhez tartozó kulcsnál kisebb, a jobbra leágazó részfában pedig annál nagyobb kulcsok foglalnak helyet.

bináris kód (binary code) – olyan →kód, amely az ábrázolandó fogalmat csak két különböző jellel v. ezek kombinációjával jeleníti meg (→binárisan kódolt decimális kód, →EBCDIC-kód, bikvináris kód, →ASCII-kód stb.).

bináris komplementum (binary complement) – olyan komplementum, amit a bináris számrendszerben dolgozó számítógépeknél a negatív számok ábrázolására használnak (→komplemens kód).

bináris mintázat (binary pattern) – bináris számjegyek rendezett sorozata, amit nem (v. nemcsak) bináris számként, hanem a bitekhez v. bitcsoportokhoz rendelt információvektorként értelmeznek (bizonyos szempontból az →EBCDIC-kód v. az →ASCII-kód is ~).

bináris-oktális konverzió (binary-to-octal conversion) – kettes számrendszerben ábrázolt számok átalakítása →oktális számábrázolásba. Mivel $8 = 2^3$, minden kettes számrendszerben kifejezett szám egyben oktális számnak is tekinthető, ugyanis minden hármas bitcsoport oktális jegyet képvisel:

000	0	100	4
001	1	101	5
010	2	110	6
011	3	111	7

Bináris számot tehát úgy írnak át oktális számmá, hogy a bináris ponttól balra és jobbra a biteket hármas csoportokba fogják össze, és minden ilyen csoportot a megfelelő oktális jeggyel helyettesítenek.

bináris összeadás (binary addition) – →bináris ábrázolású operandusokkal végzett összeadás, a következő táblázat alapján:

+	0	1
0	0	1
1	1	10

bináris pont (binary point) – a tizedespont megfelelője a kettes számrendszerben: az egész jegyeket választja el a törtrész jegyeitől.

bináris szám (binary number) – kettes számrendszerben ábrázolt szám.

bináris számláló (binary counter) – impulzusok számlálására alkalmas olyan számláló, amely az impulzusszámot bináris számábrázolásban képezi.

bináris szorzás (binary multiplication) – →bináris ábrázolású operandusokkal végzett szorzás, a következő táblázat alapján:

·	0	1
0	0	0
1	0	1

bináris tárkép forma (memory image format) – programok bináris formája, amelyben azok a →fordítás és →kapcsolás után valamely →háttértárban megtalálhatók. A →tárgyprogramokhoz tartozó tulajdonképpeni bináris szövegen kívül tartalmazza mindazokat a járulékos információkat, amelyek az →átfedési szerkezetre, ill. a →betöltésre vonatkoznak.

BIOS (Basic Input-Output System) – az IBM →személyi számítógépein →permanens memóriába égetett kiszolgáló →szubrutinok gyűjteménye. Ezek látják el a rendszer alapvető feladatait, s közvetlen kapcsolatban állnak a →hardverrel és a →perifériával. A ~ szolgáltatásai →megszakítások útján aktivizálhatók.

bipoláris kódolás (bipolar coding) – bináris jelsorozat olyan átviteli módszere, ahol a bináris 0 küldése impulzushiányt, a bináris 1-es küldése pedig impulzust jelent. Előjele minden 1-es küldésekor megváltozik, ezért tkp. háromértékű jel.

bit (bit) – 1. Az információmennyiség, ill. az entrópia mértékegysége a kommunikáció statisztikai elméletében. – 2. →Bináris ábrázolásban használt számjegy, értéke 1 v. 0.

bitbeültetés (bit stuffing) – különböző sebességű adatvonalak és egy saját sebességű →multiplexcsatorna illesztését megvalósító módszer. A multiplexcsatorna frekvenciája valamivel nagyobb, mint a kapcsolódó legnagyobb sebességű adatvonalé. A módszer tkp. „üres" bitek behelyezését jelenti a bitáramlásba, előre meghatározott időpontokban és helyeken. A töltőbit megjelenését a vételi oldal felé jelezni kell, hogy az eltávolításáról gondoskodni tudjon.

bitminta (bit pattern) – megfelelő hosszúságú bináris vektor, amelynek jól meghatározott szerepe (értelmezése) van bizonyos folyamatban v. feldolgozásban. ~nak tekinthető pl. a →gépi utasítás →címrészének tartalma a →léptetéseknél, de alapjában véve ~ minden adat bináris kódja is a számítógép belső ábrázolásában.

bitpárhuzamos átvitel (bit-parallel transmission) – adatátvitelben használható olyan módszer, amelynél az egyes karakterek bitjeit egyszerre, de magukat a karaktereket egymás után viszik át, ezért párhuzamos átvitelnek is nevezik. A ~hez az egyszerre átvinni kívánt bitek számával megegyező számú elemi csatornára van szükség, ezért nagyobb távolságú átvitelre csak ritkán használják.

bitsorkódoló (scrambler) – →digitális adatátvitelnél a →modemekben használt olyan kódolókészülék, amely nyilvánvalóan véletlen bitsorrendet hoz létre: a bejövő bináris jelsorozatot úgy kódolja, hogy az ismétlődő jelsorozatok előfordulása valószínűtlenné válik. Segítségével az adatátviteli vonalon keresztül küldött adatokban az ártalmas ismétlődő minták elkerülhetők (bár így is előfordulhatnak, de kis valószínűséggel). Az adási oldal ~jának a vételi oldalon bitsordekódoló felel meg.

bitsoros átvitel (bit-serial transmission) – adatátvitelben használható olyan módszer, amelynél a karaktereket és az azokat felépítő biteket egymás után viszik át. Háromféle módon valósítható meg: →aszinkron adatátvitellel, →szinkron adatátvitellel és a kettő kombinációjának tekinthető impulzusfolyam-adatátvitellel.

bitsűrűség (bit density) – →tárolási sűrűség

bitszelet (bit slice) – elemi →mikroprocesszor, amely korlátozott számú (rendszerint 2, 4, 8, 16) →bittel dolgozik, és olyan csatlakozási felülettel rendelkezik, amellyel több hasonló egység egymáshoz kapcsolha-

tó (→bitszelet-architektúra). Ilyen módon igen nagy teljesítményű →processzorok, →központi vezérlőegységek stb. építhetők.

bitszelet-architektúra (bit-slice architecture) – olyan →architektúra, amelynél a →központi vezérlőegység több →bitszelet összekapcsolása révén jön létre, s így a →mikroprogramozás igényeit kielégítő, →parallel működésű számítógép építhető.

bittérképezés (bit mapping) – a →képernyőn megjelenített kép kezelésére használt technika. Lényege, hogy a képernyőn megjelenített pontokat a →memóriában tárolt →bitekre leképezve tartalmazza, amelyek az illető pontok jellemzőit is rögzítik (fény, szín stb.). A megjelenítő áramkör a tárat ciklikusan kiolvassa és a képernyőre küldi, ahol fénypontokká alakítja át (ezek a →képpontok).

bizonylat (document) – valamely feldolgozandó eseményt rögzítő, rendszerint kézírásos adathordozó, amit a felhasználó tölt ki forrásadatokkal. Ezeket közbenső adathordozó (→lyukkártya, →lyukszalag stb.) segítségével lehet a számítógépbe küldeni, v. közvetlenül a ~ is lehet beviteli adathordozó, de akkor →szabványírással v. →mágneses írással kell készülnie.

bizonylatolvasó (document reader) – számítógépek →bemeneti egységeként használatos optikai v. mágneses karakterolvasó, amely eredeti →bizonylatokról olvassa be a feldolgozandó adatokat.

b/k egység (i/o unit) – egyesített →bemeneti és →kimeneti egység. Olyan berendezés, amely az adatoknak a számítógépbe való bevitelére és az eredményadatok kivitelére egyaránt alkalmas, pl. →konzolírógép, képernyős →terminál stb.

b/k gyűjtősín (i/o bus) – a legtöbb kisszámítógép szabványos tartozéka a →b/k műveletek végrehajtására. Legfontosabb vonalai az eszközök címvonalai, a parancsvonalak, az adatvonalak, a megszakítási vonalak. A perifériavezérlő egységek az adatvonalakra párhuzamosan csatlakoznak, és állandóan figyelik a címvonalat. A →központi vezérlőegység a periferikus mű címét a címvonalra helyezi, és aktivizálja a parancsvonalat. A perifériavezérlő felismeri saját címét, elfogadja és végrehajtja a kapott parancsot, ami lehet az eszköz állapotának a vizsgálata, a mechanizmus működésének vezérlése, megszakítás kezdeményezése, egy szó küldése v. fogadása stb.

b/k modul (i/o module) – a →moduláris programozás elvei alapján előállított program olyan aktív →modulja, amely a program és a külvilág kapcsolatát teremti meg, vagyis a →b/k műveleteket végrehajtja. Bár jelenléte a programokban nem feltétlenül szükséges, használata mégis előnyös, sőt ajánlatos mindegyik →[adat]állományhoz egy-egy külön ~t rendelni. Ezáltal az esetleges későbbi adatszervezési változtatás csupán a megfelelő ~t érinti.

b/k művelet (i/o operation) – az adatok bevitelét és/vagy kivitelét végző műveletek rövidített elnevezése és igen gyakori jelölési formája a számítástechnikai szakirodalomban.

b/k processzor (i/o processor) – számítógéprendszerben a →b/k műveleteket vezérlő speciális →processzor, amely a →központi vezérlőegység felügyelete alatt, de tőle függetlenül fejti ki tevékenységét. Miután a b/k műveletre vonatkozó információkat megkapta, a ~ a központi vezérlőegységgel szimultán, tőle függetlenül működhet, felszabadítva azt más folyamatok számára (→adatcsatorna).

blokk (block) – 1. Általános értelmezésben annak az →operatív memóriaterületnek az elnevezése, amit állandó jelleggel v. időlegesen egy adathalmaz számára kiosztottak. Kiemelt jelentősége van a →dinamikus memóriakiosztást megvalósító stratégiákban, ahol a dinamikus adatstruktúra (→programblokk, →lap, →szegmens) előállításakor (betöltésekor) egy ~ot utalnak ki, és egy ~ot szabadítanak fel az adatstruktúra megszüntetésekor. – 2. A →fizikai bejegyzéssel rokon értelemben használt elnevezése annak az adatmennyiségnek, amelyet egyetlen →b/k művelet segítségével töltenek be az →adathordozóról az →operatív memóriába v. onnan vissza az adathordozóra. Egyes adathordozóknál a ~ hosszát technikai jellemzők határozzák meg (pl. →lyukkártyánál), másoknál viszont csak hatékonysági meggondolások (pl. a →mágnesszalagnál). →Logikai [adat]állományoknál a ~ egy v. több →logikai rekordot jelent (→blokkolás). – 3. →programblokk

blokkdiagram (block diagram) – →folyamatábra

blokkellenőrző karakter (block check character) – a →ciklikus kóddal megvalósított adatátvitelben, a →polinom-ellenőrzésben használt speciális, karakterméretű ellenőrző bitsorozat. Az átvitelre szánt blokk bitjeivel képzett polinom és egy generátor polinom között a modulo-2 algebra szabályai szerint műveleteket végeznek, s az eredményt az eredeti blokk után viszik az átviteli vonalra. Vételnél megismétlik ezeket a műveleteket, s az eredményt összehasonlítják a ~rel. Különbség észlelésénél megismétlik a hibásnak bizonyult blokk átvitelét.

blokk-kizáró karakter (block ignore character) – →logikai [adat]állományok feldolgozásánál használt speciális karakter, annak jelzésére, hogy az illető →fizikai bejegyzés – bár fizikailag jelen van az adathordozón – nem része az [adat]állománynak. Lényegében a logikai blokktörlés eszköze.

blokk-köz (gap) – →mágnesszalagon a →fizikai bejegyzéseket (blokkokat) elválasztó zóna, ahol a mágnesszalag állva marad a készülékben, ill. ahonnan minden olvasó v. író művelet elindul. →Hajlékonylemezen két blokk közti rész, ahova nem szabványos lemezeléréssel írni, ill. visszaolvasni lehet. Ide írnak pl. a másolásvédelmi rendszerek is.

blokkolás (blocking) – mágneses →adathordozók hatékony kihasználását biztosító tárolási módszer. A jó helykihasználás érdekében a →fizikai bejegyzéseket több →logikai rekordból építik fel. A →mágnesszalagnál az így előállított blokk hosszát úgy méretezik, hogy a fizikai bejegyzések között üresen maradó helyek (→blokk-köz) az egész tárolónak csak kis hányadát jelentsék. →Mágneslemeznél a blokkok hosszát technikai jellemzők is befolyásolják (→cilinder, →sáv). Egyetlen beviteli utasítás hatására egész blokk kerül az →operatív memóriába (→puffer), ahonnan az igényelt rekord az →[adat]állomány-kezelő rendszer segítségével kerül a program munkaterületére, v. maga a felhasználói program gondoskodik erről. Az egyetlen fizikai bejegyzésbe tömörített logikai rekordok száma a →blokkolási tényező, meghatározása az [adat]állományok tervezésének egyik fontos művelete.

blokkolási tényező (blocking factor) – a →blokkolás jellemzője: az egyetlen →fizikai bejegyzésbe tömörített →logikai rekordok számát határozza meg.

blokkstruktúra (block structure) – egyes magas szintű programozási nyelvekre (→ALGOL, →PL/1, →Pascal, →Ada stb.) jellemző szerkezeti felépítés. A programok egymástól független, önálló, csak logikai kapcsolatban álló részekből, →programblokkokból épülnek fel. Alapvető jelentősége, hogy lehetővé teszi a →dinamikus memóriakiosztást, vagyis a kiosztott tárterület automatikusan felszabadul és újra kiosztható, amikor az adott blokk végrehajtása befejeződik.

blokkvége karakter (end-of-block character) – →blokkok (→fizikai bejegyzések) be-, ill. kivitelében használt →vezérlőkarakter (EOB).

Boole-[adat]típus (Boolean [data] type) – →erősen típusos nyelvekben használható →előre definiált, →sorszámozott [adat]típus logikai értékek kezelésére. Értékkészletéhez a False és True →literálok tartoznak, megengedett műveletei az értékadás, a →konjunkció, a →diszjunkció, ill. az összehasonlítások. Megjegyzendő, hogy minden összehasonlítás ~ú értéket ad eredményül. A ~ egy kevésbé pontosan definiált változata a klasszikus programozási nyelvekben – így pl. a →FORTRAN-ban – is megtalálható: a logikai [adat]típus.

Boole-algebra (Boolean algebra) – a szimbolikus logika fogalmait kifejező matematikai formalizmus, amelyet G. Boole angol matematikus javasolt 1854-ben. 1948-ban a Massachusetts Institute of Technology kutatója, C. E. Shannon javasolta alkalmazását a jelfogós kapcsolóáramkörök tervezésénél felmerülő problémák megoldására.

A számítástechnikában a kételemű ~t használják, amit a Boole-változók halmazán értelmeznek. Minden ilyen változó csak két értéket vehet fel, „igaz" (1) v. „hamis" (0) jelöléssel. (Általánosabb ~kban 2^m, m>1 elem létezik.)

A ~ban a következő műveleteket definiálják: →logikai szorzás (jelei: $\cdot$, $\wedge$, ÉS), →logikai összeadás (jelei: +, $\vee$, VAGY), →negáció (jelei: –, $\neg$, NEM). Ezekkel a műveletekkel a következő fontosabb összefüggések fejezhetők ki:

$x+x=x \quad x+1=1 \quad x+0=x \quad x+\bar{x}=1$

$x \cdot x=x \quad x \cdot 1=x \quad x \cdot 0=0 \quad x \cdot \bar{x}=0$

$$\left.\begin{array}{l} x+y=y+x \\ x \cdot y=y \cdot x \end{array}\right\} \text{kommutativitás}$$

$$\left.\begin{array}{l} (x+y)+z=x+y+z \\ (x \cdot y) \cdot z=x \cdot y \cdot z \end{array}\right\} \text{asszociativitás}$$

$$\left.\begin{array}{l} x \cdot (y+z)=x \cdot y+x \cdot z \\ x+(y \cdot z)=(x+y) \cdot (x+z) \end{array}\right\} \text{disztributivitás}$$

$$\left.\begin{array}{l} (\overline{x+y})=\bar{x} \cdot \bar{y} \\ (\overline{x \cdot y})=\bar{x}+\bar{y} \end{array}\right\} \text{de Morgan tételei}$$

Boole-függvény (Boolean function) – →logikai függvény

botkormány (joystick) – →személyi számítógépekhez kapcsolódó, a →billentyűzet →nyílbillentyűinek megfelelő funkciókat ellátó vezérlőeszköz. Segítségével a →helyőr képernyőn elfoglalt helyét lehet változtatni. (Elsősorban számítógépes játékoknál használják.)

bővítőkártya (extension card) – →személyi számítógép →alaplapjához dugaszolással csatlakoztatható áramköri lap, amellyel az alapkiépítés teljesítményét növelni lehet. A ~ általában v. újabb funkciókat valósít meg (minőségi bővítés), v. már meglévő funkciók kapacitását növeli (mennyiségi bővítés).

bpi (bits per inch) – a →tárolási sűrűség mértékegysége (bit hüvelykenként).

bps (bits per second) – →adatátviteli sebesség mértékegysége (bit másodpercenként).

buborékrendezés (bubble sort) – →helycserés rendezés

busz (bus) – →sínrendszer

C

C – 1. →Hexadecimális ábrázolás jelkészletének eleme, amely a decimális 12 értéknek felel meg (bináris megfelelője: 1100). – 2. Magas szintű →programozási nyelv, amelyet Dennis Ritchie fejlesztett ki a →DEC PDP-11 számítógépre írt →UNIX →operációs rendszer részére, majd megvalósították számos más rendszeren is, többek között az IBM System/370, Honeywell 6000, ill. Interdata 8/32 részére. Modern vezérlési struktúrákkal, adatszerkezetekkel, gazdag operátorkészlettel rendelkező programozási nyelv. Hatékony eszközökkel rendelkezik a modern számítógépek összes lehetőségeinek kihasználására, ezért alkalmas operációs rendszerek, rendszerprogramok stb. írására. (Így pl. az UNIX operációs rendszer több mint 13 000 programsorából csupán 800 készült →asszembler nyelven, sőt maga a ~ nyelv →fordítóprogramja is ezen a nyelven készült.) Hatékony →hordozhatóságot is biztosít, nagyon kis befektetéssel implementálható bármelyik számítógéprendszeren. A ~ jellegzetes példája az egy nyelv köré épülő →programozási környezet megvalósításának, ahol az operációs rendszer, a rendszerprogramok és a fejlesztési programok ugyanazt a programozási nyelvet használhatják. Legújabb C++ változata már az →objektumorientált programozást is lehetővé teszi.

CDL (Compiler Description Language) – magas szintű →programozási nyelv, amelyet rendszerprogramok, elsősorban →fordítóprogramok írására fejlesztettek ki, abból a szemléletből kiindulva, hogy a programozási nyelvek →szintaxisa a →Backus-Naur Forma formalizmusával előnyösen leírható. A ~ központi fogalma a szabály, amelynek más nyelvekben az eljárás felel meg:

<meghatározandó-kategória>::= kifejezés

vagyis a bal oldalon szerepel a szabály neve, a jobb oldalon pedig az alternatívák sorozata.

A ~ építőkövei a →makroutasítások: ezekben fogalmazzák meg az elemi tevékenységeket, vizsgálatokat, ezeken keresztül érhetők el a számítógép →erőforrásai, a makroutasításokat a célgép →asszembler nyelvén kell megírni. Továbbfejlesztett változata a CDL2.

célszámítógép (special-purpose computer) – jól meghatározott feladat v. feladatkör megoldására tervezett számítógép, amely messzemenően figyelembe veszi az illető feladatkör jellegzetességeit. Szemben az általános célú számítógéppel, a ~ csak azokat az →erőforrásokat tartalmazza, amelyekre az adott feladatkörben ténylegesen szükség lehet, azokat viszont a legmagasabb szinten. A ~ elterjedése elsősorban a →mikroprocesszorok megjelenésének és egyre nagyobb megbízhatóságának köszönhető.

centralizált feldolgozás (centralized processing) – olyan feldolgozási mód, amelynél a felhasználók (v. azok egy csoportja) minden feldolgozási igényét központilag elhelyezett számítógép →erőforrásaihoz való hozzáféréssel lehet kiszolgálni (szemben az →elosztott feldolgozással).

CGA (Colour Graphic Adapter) – →szöveges módban és →grafikus módban egyaránt működő →képernyőadapter. Egyszerű →felbontásnál (320x200 →képpont) a képernyőn egyszerre 4 szín lehet (egy alapszín és három másik), dupla felbontásnál (640x200 képpont) csak két szín lehetséges.

CHDL (Computer Hardware Description Language) – olyan magas szintű →programozási nyelv, amely a →hardver leírására alkalmas. Az ilyen típusú nyelvek száma rendkívül nagy, ezek közül a legismertebbek: CDL (Computer Design Language – számítógéptervező nyelv); DDL (Digital Design Language – digitális tervező nyelv); AHPL (A Hardware Programming Language – egy hardverprogramozási nyelv); ISP (Instruction Set Processor – utasításkészlet-processzor); PMS (Processor-Memory-Switch – processzor-tárkapcsoló) stb.

CHILL (Ccitt HIgh Level Language) – a Nemzetközi Távíró és Távbeszélő Tanácsadó Bizottság (CCITT) által 1980-ban elfogadott magas szintű →programozási nyelv, az →erősen típusos nyelvek egyik legfontosabb képviselője. Elsősorban a programvezérelt távközlési kapcsolórendszerek fejlesztésére alkalmas, de modern adat- és vezérlőstruktúrái révén jól használható →operációs rendszerek, nagyméretű vezérlőrendszerek, magas szintű ember-gép kapcsolat megvalósítására is. Elsőrendű célkitűzése a magas fokú

megbízhatóság megteremtése, gazdag és hatékony eszköztárat tartalmaz →konkurens folyamatok vezérlésére, →kivételek biztonságos kezelésére stb. Szekvenciális része a →Modula szerkezeteire épül, de igen sok egyéni megoldás is jellemzi (pl. a →külön fordítás, az →információrejtés stb.). Kommunikációs és szinkronizálási eszközei révén elsősorban az elosztott kommunikációs távközlési rendszerek fejlesztésére alkalmas.

Chomsky-hierarchia (Chomsky-hierarchy) – a formális →nyelvtanok osztályozása helyettesítési szabályok alapján. Noam Chomsky a nyelvtanokat négy osztályba sorolta a helyettesítési szabályok formájára tett megszorítások alapján. (Ezek megértéséhez tudnunk kell, hogy T*-gal jelölik a T ábécé elemeiből alkotható összes véges hosszúságú jelsorozatot, az üres jelsorozatot is beleértve; T+-szal jelölik a $T*-\lambda$ halmazt, vagyis az üres jelsorozat nélküli, a T ábécé elemeiből alkotható összes véges hosszúságú jelsorozatot.)

1) 0 típusú nyelvtan (mondatszerkezetű nyelvtan): a legáltalánosabb; helyettesítési szabályaira semmilyen megszorítás nem vonatkozik. A szabályok formája:

$\alpha \rightarrow \beta$, ahol $\alpha \in N \cup T*$ és $\beta \in T*$

vagyis a szabály bal oldalán →terminális szimbólumokból és →nemterminális szimbólumokból alkotott tetszőleges nem üres szöveg áll, míg a jobb oldalán megengedett az üres jelsorozat is.

2) 1 típusú nyelvtan (nemcsökkentő v. környezetfüggő nyelvtan): helyettesítési szabályainak az a tulajdonsága, hogy

$\tau\alpha\delta \rightarrow \tau\beta\delta$,
ahol $\tau,\delta \in N \cup T*$ és $\alpha,\beta \in N \cup T+$

vagyis a szabályok legalább annyi szimbólumot tartalmaznak a jobb oldalon, mint a bal oldalon, más szóval $l(x) \leq l(y)$.

3) 2 típusú nyelvtan (környezetfüggetlen nyelvtan): helyettesítési szabályainak az a tulajdonsága, hogy

$A \rightarrow \alpha$, ahol $A \in N$ és $\alpha \in N \cup T*$

vagyis a szabályok bal oldalán csak önmagukban álló nemterminális szimbólumok fordulhatnak elő. A →programozási nyelvek elméletében és gyakorlatában ez a nyelvosztály játssza a legfontosabb szerepet.

4) 3 típusú nyelvtan (reguláris nyelvtan) helyettesítési szabályainak formája:

$A \rightarrow \alpha$

$A \rightarrow B\,\alpha$ (bal lineáris)

$A \rightarrow \alpha\,B$ (jobb lineáris),
ahol $A,B \in N$ és $\alpha \in T$

A ~ szerint a nyelvtanokat úgy származtatják egymásból, hogy a helyettesítési szabályok formájára újabb és újabb megszorításokat tesznek. Így nyilvánvaló, hogy minden reguláris nyelvtan egyben környezetfüggetlen nyelvtan is, továbbá a környezetfüggetlen nyelvtan környezetfüggő is, végül a környezetfüggő nyelvtan mondatszerkezetű nyelvtan is. A ~ban tkp. minden nyelvtani osztály a közvetlenül felette található osztály részhalmaza.

A nyelveket az őket generáló nyelvtan típusa határozza meg: az i típusú nyelvet i típusú nyelvtannal lehet generálni, de i+1 típusúval már nem (i=0,1,2).

ciklikus kód (cyclic code) – az →adatátviteli hálózatokban legjobban elterjedt →kód, amely a véletlen és a csomós hibák hatékony jelzésére és javítására egyaránt alkalmas. (A csomós hibát az jellemzi, hogy az első és utolsó meghibásodott csatornaszimbólum között elhelyezkedő szimbólumok meghibásodási valószínűsége nagyobb, mint ezeken kívül; a gyakorlatban a hibacsomók főleg 2-3 egymás melletti bit azonosra változását jelentik.)

A ~ lényege az, hogy az átvinni kívánt blokk bitjeit egy polinom együtthatóiként fogják fel, s így velük aritmetikai műveletek végezhetők. Átvitel során minden blokkhoz hozzárendelnek egy generátorpolinomot úgy, hogy a két polinom között a modulo-2 algebra szabályai szerint műveleteket végeznek. Eredményként az eredeti blokk után ellenőrző bitek keletkeznek (→blokkellenőrző karakter), s ezeket viszik utoljára az átviteli vonalra. A vevőoldalon a műveleteket megismétlik, és az eredményt összevetik az ellenőrző bitekkel.

A ~ jól alkalmazkodik a változó hosszúságú →üzenetekhez. Hatékonysága a generátorpolinom megválasztásától, a hibafajták eloszlásától és az üzenetek hosszától függ.

ciklikus léptetés (cyclic shift) – olyan →léptetés, amelynél a szót alkotó összes bitpozíciók úgy mozdulnak el, hogy az egyik szóvégen kilépett bit tartalma a másik szóvég felszabaduló bitpozíciójába lép (ezért nevezik még körkörös léptetésnek is). Így pl. a balra léptetésnél a legkisebb helyértékű bitpozíció az eredeti szó legnagyobb helyértékű pozíciójának tartalmát kapja. (Az →aritmetikai léptetéssel szemben a –nél az előjelbit is közönséges bitként vesz részt a műveletben.)

ciklikus redundancia-ellenőrzés (cyclic redundancy check) – →adatátviteli hálózatokban a vevőoldalon végzett ellenőrzés módszere, ha az átvitelre →ciklikus kódot használtak. Az ellenőrzés úgy történik, hogy a vevőoldalon megismétlik az adóoldalon végzett polinom műveleteket, s az eredményeket összehasonlítják az ugyancsak átvett ellenőrzőbitekkel. Hiba észlelésekor az átvitelt automatikusan megismételtetik.

ciklus (cycle) – bizonyos számú utasításból álló programrészlet, amelyet egymás után többször végrehajtanak. A végrehajtás megismétlését a ciklusváltozóra előre meghatározott feltétel teljesülése ellenőrzi (→ciklusutasítás).

ciklusiopás (cycle stealing) – az →operatív memória és a →periferikus művek közötti adatátvitelnek az →adatcsatorna által ellenőrzött módszere, a →közvetlen tárhozzáférés megvalósításának eszköze. Lényege az, hogy a vezérlés a csatorna oldaláról történik úgy, hogy az átvitel időtartamára a →központi vezérlőegység és az operatív memória közötti kapcsolat felfüggesztődik (a csatorna a processzor működéséből „ellopja" az átviteli ciklushoz szükséges időt).

ciklusutasítás (iteration statement) – olyan magas szintű →vezérlésátadó utasítás, amely bizonyos programrész feltételhez kötött többszöri végrehajtását teszi lehetővé. Gyakorlati megvalósításai különbözhetnek a különböző programozási nyelvekben, alapvető attribútumai viszont általánosak: a végrehajtandó programrészlet behatárolása, az ismétlést meghatározó feltétel specifikációja, a feltétel kiértékelésének momentuma.

A feltétel kiértékelése történhet a ciklus végrehajtása előtt v. után. Ciklus előtti kiértékelésnél a – lehet „WHILE feltétel DO programrészlet" alakú (a ciklusból való kilépést a feltétel hamis értéke okozza) v. „PERFORM programrészlet UNTIL feltétel" alakú (a kilépést a feltétel igaz értéke teszi lehetővé).

Ciklus utáni kiértékelésnél a – formája „DO programrészlet UNTIL feltétel" v. „REPEAT programrészlet UNTIL feltétel", s a ciklusból való kilépést a feltétel teljesülése teszi lehetővé (ennél a formánál a programrész legalább egyszer mindig végrehajtásra kerül).

A – másik megvalósítási formája a ciklusvégrehajtások számának meghatározásán alapszik: a kiértékelendő feltétel a ciklusváltozó értéke, amely egy kezdeti értékből indul, és meghatározott lépéssel haladva ér el a megadott maximális (minimális) határértékig. Ez a kilépési feltétel. Tipikus megvalósítása a →FORTRAN nyelv DO utasítása v. a →Pascal FOR utasítása stb.

cilinder (cylinder) – mozgófejes →mágneslemeznél a különböző felületeken egymás felett elhelyezkedő és így az író-olvasó fejek elmozdítása nélkül elérhető →sávok összessége. A –t alkotó sávok száma típusonként változó, az →adathordozó technikai jellemzője. (A magyar szakirodalomban nevezik még palástnak is.)

cilinderindex (cylinder index) – →indexszekvenciális [adat]állománynál a →sávindexnél magasabb indexszint neve. A – az egyes →cilindereken található legmagasabb értékű →kulcsot tartalmazza, és az adott cilinder sávindexére mutat.

cím (address) – 1. Olyan szám- v. jelkombináció, amelynek segítségével az →operatív memória rekeszei, a →regiszterek v. bizonyos →adathordozók zónái egyértelműen, közvetlenül és megismételhető módon lokalizálhatók, a tárolt információk hozzáférhetők. Az operatív memóriával kapcsolatban a –ek több típusa ismeretes: →abszolút cím, →számított cím, →báziscím, →indexelt cím, →indirekt cím, →eltolás, →virtuális cím stb. – 2. Az →adatátviteli hálózatok értelmezésében egy v. több →karakter, amely egyértelműen meghatároz egy-egy →adatállomást (adót v. vevőt).

címábrázolás (address representation) – az →operatív memória →abszolút címeinek megjelenítési formája a →gépi utasítások →címrészében. A közvetlen –nál a címrész

az abszolút cím értékének bináris formája. Előnye, hogy a →központi vezérlőegység közvetlenül értelmezheti, hátránya viszont, hogy a nagy kapacitású memóriáknál igen helyigényes és változó hosszúságú. A közvetett ~nál a címrész csupán valamilyen alaphoz viszonyított távolságot jelöl (→relatív cím, →báziscím, →virtuális cím). Előnye, hogy kis helyet igényel, és az utasítás hossza sem függ az ábrázolt cím(ek)től, viszont a tényleges felhasználása előtt →címátalakítást, ill. →címszámítást tesz szükségessé.

címátalakítás (address mapping) – a →virtuális memória kezelésének alapvető művelete, amelynek során az →operációs rendszer a végrehajtás alatt álló program logikai (virtuális) címeit az →operatív memória fizikai címeire leképzi. A ~ általában címtérkép segítségével valósul meg, amelynek elemeit egy-egy logikai-fizikai címpáros alkotja. Ilyen formában azonban a címtérkép irreálisan nagy volna, ezért a módszert módosítva szokták használni.

A →lapkezelésnél minden logikai cím kétkomponensű virtuális cím: c=(p,w), ahol p az adott lap azonosító száma, w pedig az operandus lapon belüli relatív címe. A címtérképet laptérkép váltja fel, amely a lapok azonosítója és a →betöltési cím között teremt kapcsolatot.

A →szegmentálásnál a logikai cím szintén kétkomponensű virtuális cím: c=(s,w), ahol s az adott szegmens azonosítója, w pedig a szegmensen belüli relatív cím.

Tovább finomítva a módszert, a logikai címet három komponensből álló virtuális címként kezelik: c=(s,p,w), ahol s az adott szegmens, p a szegmenshez tartozó lap azonosító száma, w a lapon belüli relatív cím. A ~hoz szegmenstérkép és laptérkép tartozik. Minthogy a leképzéshez két táblázatban is keresési műveletet kell végrehajtani, a hozzáférési idő nagyjából háromszorossá növekedhet, ezen csak megfelelő →asszociatív memória közbeiktatásával lehet segíteni.

címáthelyezés (address translation) – az →áthelyezhető bináris formában kódolt programegységekben használt címek átalakítása a →kapcsolás, a →betöltés és a végrehajtás alatt. A →fordítás nyomán létrejött →tárgyprogramokban a →modulok kezdetéhez viszonyított →relatív címeket használnak. Kapcsoláskor ezeket úgy alakítják át, hogy minden cím a létrejövő →szegmens kezdetéhez viszonyuljon.

címezhetőség (addressability) – a számítógépek működésének az a fontos alapelve, hogy az utasításokba nem magukat az adatokat (operandusokat) írják be, amelyekkel az adott műveletet el kell végeznie, hanem azoknak →operatív memóriabeli →címeit. A →központi vezérlőegység az utasításokban szereplő címek alapján lehívja a memóriából a megfelelő adatokat, végrehajtja ezekkel az előírt műveletet, az eredményt pedig esetleg visszahelyezi a memóriába. A ~ elve alapján a feldolgozó programok és a feldolgozandó adatok egyaránt az operatív memóriában tárolhatók. Ugyancsak a ~ teszi lehetővé az →iterációs eljárások programozhatóságát.

címke (label) – 1. →Programozási nyelvekben az utasítások azonosítására, a hivatkozások jelölésére használt jelkombináció, amelynek segítségével a →vezérlésátadó utasítások megvalósítják a programelágazásokat. Szintaktikai formája a különböző nyelvekben más és más lehet (csak decimális számjegyek, mint pl. a →FORTRAN-ban v. →Pascalban, betűk és számjegyek tetszőleges kombinációja, mint pl. a →COBOL-ban v. az →azonosítókra vonatkozó általános szabályok alapján képzett formátumok a programozási nyelvek nagy többségében). A ~ket a →fordítóprogramok alakítják át a →tárgyprogram címeire. – 2. →Adathordozók szabványos szervezésében használt információs blokk (→szabványos címke, →[adat]állománykezdőcímke, →[adat]állomány-zárócímke stb.).

címkekonstans (label constant) – a →PL/1 programozási nyelvben a →címkét nevezik ~nak is, érzékeltetve azt, hogy ez a nyelv ismeri a →címkeváltozó fogalmát. A ~ tehát olyan azonosító, amelyet utasítások elé írnak prefixumként, hogy az őt követő utasításra hivatkozni lehessen a →vezérlésátadó utasításokban. A címkeváltozók →aktuális értékei ~ok.

címkeváltozó (label variable) – a →PL/1 nyelvben érvényes →[adat]típushoz tartozó olyan →változó, amelynek értékei →címkekonstansok. Lehetővé teszi, hogy a →vezérlésátadó utasítások a ~ aktuális értékével jel-

zett utasításra hivatkozzanak. A ~ a vezérlésátadó utasításokon és a →deklarációkon kívül csak értékadó utasításokban szerepelhet.

címkifejezés (address expression) – →asszembler nyelven kódolt programokban operandusok →címének meghatározására használatos egyszerű aritmetikai kifejezés. Két alapvető formája az →abszolút kifejezés, ill. az →áthelyezhető kifejezés.

címkonstans (address constant) – →asszembler nyelvekben használatos olyan konstans, amelyhez meghatározott memóriarekesz →címe van hozzárendelve, s amellyel speciális műveletek végezhetők. Elsőrendű szerepe van az →átfedési szerkezetek forrásnyelvű megvalósításában (→szegmensek betöltésében, →belépési pontok meghatározásában és a rájuk történő hivatkozásokban stb.), ugyanis ezeket az asszembler nyelvben szimbolikus formában lehet kezelni. A ~okról az asszemblernek külön listát kell készítenie, mert az utólagos feldolgozást a →kapcsolóprogram, ill. a →betöltőprogram végzi el.

címlista szerinti adatútképzés (directory routing) – az →adatútképzés egyik módszere. A hálózat csomópontjaiban címtáblázatok állnak rendelkezésre, amelyekből meghatározható, hogy megadott rendeltetésű →adatcsomagoknak melyik összeköttetésen kell továbbjutniuk. A címtáblázatok tartalékként javasolt összeköttetéseket is tartalmazhatnak.

címmódosítás (address modification) – valamely →gépi utasítás →címrészét érintő módosítás. Három alapvető változata ismeretes. Az →indexregiszteres ~nál a tényleges címet az →eltolás és egy (v. több) indexregiszter tartalmának összege adja. A →bázisregiszteres ~nál az abszolút címet az eltolás és a bázisregiszter tartalma összegeként számítják. Az eltolást érintő ~nál a gépi utasítás címrészét adatként kezelve módosítják, s utána hajtják végre a megváltoztatott utasítást. A ~ végrehajtható a három módszer vegyes felhasználásával is.

cím nélküli utasítás (addressless instruction) – olyan →gépi utasítás, amelynél v. hiányzik a →címrész, v. annak más jelentősége van mint operanduscím (pl. a →léptetést végző utasításoknál). Ugyancsak ~okat használnak a →veremműveletek, amelyeknél az elérhető operandus(ok) mindig a →verem csúcsa(i), s csak a művelet kódjára van szükség az utasításokban. Ez az utóbbi szervezési módszer elsősorban a →mikroprocesszorokra jellemző.

címregiszter (address register) – 1. A →központi vezérlőegység speciális →regisztere, amely az éppen aktuális utasítás operandusának az →abszolút címét tartalmazza. – 2. Az →operatív memória vezérlőegységének regisztere, amelynek minden →memóriaciklus elején azt az abszolút címet kell tartalmaznia, ahova adatot kell beírni, ill. ahonnan a kiolvasás történik.

címrendezés (address sorting) – címezhető →adathordozón (pl. →mágneslemezen) tárolt →logikai [adat]állományok →rendezésére alkalmas eljárás. Lényege, hogy az →operatív memóriában összeállítanak egy olyan táblázatot, amelybe a rendezendő →rekordok →fizikai címeit és a rendezés alapjául szolgáló adatmező(ke)t veszik fel. A táblázat elemeit valamilyen →belső rendezési eljárással átrendezik, majd ezen keresztül, közvetett címzéssel férnek az illető rekordokhoz. Ezzel a módszerrel az [adat]állomány rendezettnek tekinthető a felhasználás szempontjából anélkül, hogy a nagy adattömeg tényleges mozgatására szükség volna. A rendezett (index)táblázatok maguk is tárolhatók [adat]állomány formájában. Ezt a módszert elsősorban az →adatbázisoknál használják, ahol az [adat]állományokat rendszerint igen gyakran kell különböző szempontok szerint átrendezni.

címrész (address part) – a →gépi kód operandusrésze, vagyis az a rész, amely a műveletben részt vevő operandus →címét tartalmazza. Aritmetikai, logikai és átviteli utasításoknál a ~ általában a programegység kezdetéhez viszonyított →relatív cím (→eltolás); egyes utasításoknál viszont a ~ nem tényleges cím, hanem a művelet végrehajtásához szükséges járulékos információk kódja (pl. a →léptetésnél).

címsín (address bus) – →sínrendszer

címszámítás (address computing) – a →központi vezérlőegység által végzett művelet, amelynek során a →gépi utasításban található címinformációk alapján előállítja az →operatív memória megfelelő →abszolút címét. Bázisregiszteres címzésnél az abszolút címet a →bázisregiszter tartalmának és a

→címrészben található →eltolásnak az összege adja. →Indexelt címzésnél az indexregiszter tartalmát adják hozzá az eltoláshoz. →Indirekt címzésnél az első – nem az operandus közvetlen címe, hanem az az abszolút cím, ahol az operandus tényleges címe található. A címképzési módszerek vegyes használatánál pontos szabályok rögzítik a műveletek sorrendjét.

cím szerinti hivatkozás (call by address) – →eljárások hívásánál használt paraméterátadási módszer. Lényege az, hogy az eljárás aktivizálása előtt az →aktuális paraméterek címeit átmásolják a megfelelő →formális paraméterek számára fenntartott helyekre. Az aktuális paramétereket így →indirekt címzés útján lehet elérni. A módszer jellegzetessége, hogy a tkp. feldolgozott objektumokhoz a hívó eljárás területén rendelnek tárolóhelyet, tehát a hívott eljárás módosítja a hívó memóriarekeszeit. A – alkalmas eredmények visszaküldésére is a hívó eljárásba.

címtartomány (address space) – első megközelítésben az →operatív memóriát felépítő rekeszek →címeinek halmaza. Programozástechnikai szempontból a –nak két vetülete ismeretes. A számítógép operatív memóriája címezhető egységekből épül fel, a lehetséges címek nullától a tárkapacitás maximális értékéig növekednek, s ezek alkotják a fizikai –t. Viszont az az operatív memória, amelyben a program kódolásakor a programozó gondolkozik, programjainak címeit kialakítja, a logikai –. A logikai és a fizikai – csak a legegyszerűbb →monoprogramozásos rendszerben fedi egymást (itt sem mindig, ha figyelembe veszik az →átfedési szerkezetek megvalósíthatóságát is). →Multiprogramozásos rendszereknél lennie kell egy olyan mechanizmusnak, amely a logikai –t megfelelteti a fizikai – egy részének (→átfedési szerkezet, →áthelyezés, →virtuális memória stb.).

címzés (addressing) – valamely adat →operatív memóriabeli helyének behatárolására szolgáló eljárás, a →címezhetőség elvének gyakorlati megvalósítása. A –nél használható legfontosabb címtípusok: →abszolút cím, →báziscím, →indexelt cím, →indirekt cím, →relatív cím, →szimbolikus címzés, →virtuális címzés stb.

COBOL (COmmon Business Oriented Language) – az egyik legelterjedtebb, ügyviteli-gazdasági adatfeldolgozásra orientált, magas szintű →programozási nyelv. Nyers specifikációja 1959-ben jelent meg, majd a továbbfejlesztésének és rendszerezésének eredményeképpen 1966-ban megjelent változat érdembeni módosításokat is mutat, és szembetűnő a teljes anyag átszerkesztése is: három részre osztva tárgyalja a – történetét, a nyelv filozófiáját és specifikációját. Utolsó, ma is használt szabványosított változata a COBOL ANS-81.

Elsősorban az olyan típusú feladatok megoldására előnyös, ahol az aritmetikai műveletek mennyisége jóval kisebb, mint az adatkezelési tevékenység. Másik alapvető tulajdonsága a számítási műveletek viszonylagos szegénysége (a négy alapművelet és a hatványozás), és hiányoznak a tudományos-technikai orientációjú nyelvekre jellemző →standard függvények. Igen gazdag viszont az →[adat]állományok szervezésére és kezelésére alkalmas eszközökben. Strukturális szempontból a –-program négy főrészre tagolódik: →azonosítási, környezetleírási, adatleírási és eljárási főrészre. Ez a funkcionális tagolódás rendszerint a lefordított →tárgyprogramra is jellemző. Előnyös adottságainak köszönhetően a – gyakran az →adatbázisok →adatleíró, ill. →adatkezelő nyelvének szolgál →bázisnyelvként.

CP/M – 8 →bites →mikroszámítógépekre kifejlesztett, a maga idejében igen elterjedt →operációs rendszer. A →COBOL, →FORTRAN, →Pascal stb. nyelvekhez →fordítóprogramokat, →segédprogramokat tartalmazott, s elterjedtsége révén mintegy szabványnak is számított. Egy fejletteb változatot CP/M-86 néven 16 bites gépekre is kifejlesztettek. Az →MS-DOS megjelenésével elveszítette jelentőségét.

CS

„csak olvasható" memória (read-only memory) – →permanens memória

csapda (trap) – a végrehajtás felfüggesztése a program belső állapotában előadódó kivételes esemény nyomán. Ilyen esemény lehet az aritmetikai műveletek során felléphető →túlcsordulás v. →alulcsordulás, nem létező memóriacímre való hivatkozás, →privilegizált utasítás jelenléte a felhasználói programban, illetéktelen hozzáférési kísérlet védett memóriatartományokhoz, →ismeretlen utasítás végrehajtási kísérlete stb. Ezek kezelésére nem célszerű minden programot felkészíteni, a feladatot az →operációs rendszerre lehet bízni. A rendkívüli esemény bekövetkezésekor a felhasználói program végrehajtása felfüggesztődik, s a vezérlést az eseményt felismerő és megfelelően feldolgozó →szubrutin kapja meg. Az esemény jellegétől függően a program még visszakaphatja a vezérlést, v. a végrehajtás végérvényesen megszakad. Ezt a mechanizmust rendszerint a különböző szoftver →erőforrások aktivizálására is szokták alkalmazni. Az ismeretlen utasításkódot követő felfüggesztés után az operációs rendszer felismeri az esemény típusát, aktivizálja a kódnak megfelelő szoftver erőforrást (rendszerszubrutint), majd visszaadja a vezérlést a felfüggesztett programnak (→kiterjesztett gép, →felügyelőprogram-hívó utasítás).

csatorna (channel) – be-, ill. kimeneti műveletek vezérlésére alkalmas →processzor, amely lehetővé teszi az →operatív memória és a →periferikus művek közötti adatcsere végrehajtását a →központi vezérlőegység beavatkozása nélkül. A ~ maga is végre tud hajtani bizonyos utasításokat, →csatornaparancsokat, amelyeket →csatornaprogram formájában az operatív memóriában tároltak. A csatornaprogram végrehajtása a központi vezérlőegység kezdeményezésére indul meg, s folytatódik annak beavatkozása nélkül (természetesen a központi vezérlőegység az adatátvitel alatt bármikor beavatkozhat a ~ v. a periferikus készülék állapotának tesztelésére, sőt önkényesen leállíthatja magát az adatátvitelt is). A ~ a csatornaprogram parancsait szekvenciálisan végzi, az →adatláncolásnak, ill. →parancsláncolásnak megfelelően. Az adatátvitel végén v. bármilyen rendellenességre a ~ →megszakítás útján kérheti a központi vezérlőegység beavatkozását. Üzemmódjának megfelelően a ~ lehet →szelektorcsatorna v. →multiplexcsatorna.

csatorna-állapotszó (channel status word) – olyan információvektor, amely az →adatcsatorna jellegzetes paramétereit foglalja magában. Rendszerint a következő információkat tartalmazza: a következő →b/k művelet címe, a még átvitelre váró karakterek száma, a csatorna működését befolyásoló mutatók, a vezérelt →periferikus mű állapotát jelző információk stb. A ~ számára rögzített hely van fenntartva az →operatív memóriában, amelyet a →központi vezérlőegység és a csatorna vezérlőegysége egyaránt elér. A központi vezérlőegység a ~ban helyezi el az adatcserét vezérlő információkat, majd a csatornát elindítva, az saját maga vezérli a tulajdonképpeni adatcserét. A művelet lefolyását, a periferikus mű állapotát jellemző információkat ugyancsak a ~ban helyezik el, így a központi vezérlőegység állandó felügyelet alatt tarthatja az adatcsere lefolyását, és szükség esetén bármikor beavatkozhat.

csatornaparancs (channel command) – az →adatcsatorna által értelmezett és végrehajtott speciális utasítás. A ~ok szekvenciális végrehajtásával az adatcsatorna a →központi vezérlőegységtől függetlenül vezérelheti az adatátvitelt. A ~ot felépítő mezők a következő fontosabb információkat tartalmazzák: a parancskódmező rögzíti a végrehajtandó művelet típusát; az adatcímmező azt az első memóriacímet határozza meg, ahol az első adatszó tárolandó, ill. ahonnan az kiolvasandó; a jelzőbitmező az →adatláncolást, a →parancsláncolást, hosszjelzéselnyomást, ugrást stb. jelzi; a számlálómező az átvitelre kerülő, ill. a már átvitt →bájtok számát adja meg. Egy v. több ~ →csatornaprogramot alkot.

csatornaprogram (channel program) – speciális →gépi utasítások (→csatornaparancsok) olyan sorozata, amelynek alapján az →adatcsatorna vezérelni tudja az adatátvitelt a →központi vezérlőegység beavatkozása nélkül. A ~ utasításai egyetlen →periferikus műre vonatkoznak, és biztosítják az adatátvitelhez szükséges összes információt.

csillaghálózat (star transmission network) – az →adatátviteli hálózatok egyik legegyszerűbb formája: egyetlen feldolgozó központot kapcsol össze a köréje telepített →adatvégállomásokkal. Ezek rendszerint csak a feldolgozó központtal vannak közvetlen kapcsolatban. Néha egymás között is cserélhetnek információt a központi számítógépen keresztül, ebben a felhasználási módban azonban a számítógép csak az üzenetkapcsoló szerepét látja el. Ez a szervezési forma akkor előnyös, amikor a feldolgozó központ a geometriai központba telepíthető, az egyes adatvégállomások nem esnek túl közel egymáshoz a számítógéphez viszonyított távolságukhoz képest, és saját vonalukat az idő nagy részében lekötik.

csomag (package) – az →információrejtés konkrét megvalósítása az →Ada programozási nyelvben. Lényegében olyan programegység, amely a program többi része által használható szolgáltatásokat definiálja anélkül, hogy azok megvalósításának részleteiről konkrét információkat adna. A ~ban definiált eszközök megvalósítása így anélkül módosítható, hogy ez további módosításokat követelne a program többi részében is.

csomagkapcsolás (packet switching) – →adatátviteli hálózatokban használatos módszer, az →üzenetkapcsolás sajátos formája. ~nál az információcsere rövid →adatcsomagok formájában megy végbe. Ezeket számítógépes kezelésre tervezték, így rögzített formátumúak, az áthaladás lerövidítésére maximálva. Ha az adatcsomag hosszúságát meghaladó →üzeneteket kell továbbítani, akkor azokat adatcsomagokra osztják. A ~os hálózatnak az a célkitűzése, hogy az adatcsomagokat minimális késleltetéssel továbbítsa. Az üzenetkapcsoláshoz hasonlóan ez az eljárás sem igényli az átviteli vonalszakasz állandó lefoglalását. Az adatcsomagokat a hálózat kapcsolóközpontjai tárolják és továbbítják.

csomópont (node) – →adatátviteli hálózatok topologikus leírásában az összeköttetések találkozásának pontja. →Adathálózatoknál, de különösen a →csomagkapcsolásos hálózatoknál kapcsolóközpontot is jelent.

csomóponti kapcsoló számítógép (switching-node computer) – →csomagkapcsolásos v. →üzenetkapcsolásos adatátviteli hálózatok →csomópontjaiban elhelyezkedő kisszámítógép, amely a hálózati kapcsolóközpont szerepét tölti be. Feladata a hozzá érkező →adatcsomagok, ill. →üzenetek átmeneti tárolása és továbbítása más csomópontokba.

csonkítás (truncating) – adatok átvitelénél v. →adatkonverziók nyomán létrejöhető jelenség: az ábrázolandó érték értékes jegyei elvesznek a befogadó mező véges hosszúsága miatt. A →lebegőpontos ábrázolás rögzített, véges szóhosszúsága okozta hibát sokszor →kerekítési hibának is nevezik. A programozónak tudnia kell, hogy hosszú számításoknál a felhalmozódó ~ok az eredményben sokkal nagyobb eltéréseket eredményezhetnek, mint egyetlen kerekítés. Az ilyen típusú hibák elemzése a numerikus analízis fontos feladata.

D

D – a →hexadecimális ábrázolás jelkészletének eleme, amely a decimális 13 értéknek felel meg (bináris megfelelője: 1101).

dBase – az Ashton-Tate →szoftverház →személyi számítógépekre, →munkaállomásokra kifejlesztett relációs adatbázis-kezelő rendszer. →Szövegszerkesztőkkel, más →programozási nyelven írott programokkal is könnyen kapcsolható, kényelmes →lekérdezést biztosító →adatbázis- kezelő rendszer is egyben. Aktuális változatai a dBase III Plus és a dBase IV.

DD (Double Density) – dupla sűrűségű tárolást biztosító →hajlékonylemez típusjele (→tárolási sűrűség).

decimális-bináris konverzió (decimal-to-binary conversion) – tízes számrendszerben ábrázolt számok átalakítása kettes számrendszerbe. Az átalakításra a decimális osztást használják a következő algoritmus szerint:

1. elosztják az eredeti decimális számot 2-vel (az új számrendszer alapszámával); az osztás maradéka az átalakított kettes alapú szám 2^0 helyértékű számjegye, a hányados a következő lépés kiinduló száma;

2. elosztják az előbbi osztás hányadosát 2-vel; az osztás maradéka adja az átalakított szám következő számjegyét;

3. az előző lépést ismétlik mindaddig, amíg az osztás hányadosa nulla, amikor az osztási maradék az átalakított szám legmagasabb helyértékének számjegye.

Tizedes törtek átalakítása sorozatos szorzással végezhető el.

Minden számjegy, amely a tizedesvesszőtől balra kerül, a kettes számrendszerbeli ekvivalens egy-egy jegyét adja. A műveletet tetszőlegesen sokszor meg lehet ismételni, bár a gyakorlatban csak adott pontossági határ elérése a cél. Az első szorzásból adódó számjegy a kettes számrendszerbeli tört vesszőtől jobbra levő első számjegyét adja, majd rendre követik egymást a vesszőtől jobbra távolodva.

decimális-hexadecimális konverzió (decimal-to-hexadecimal conversion) – tízes számrendszerben ábrázolt számok átalakítása →hexadecimális ábrázolásba. Az egész számok átalakítása a 16-tal való sorozatos osztással, a törteké sorozatos szorzással végezhető el (a →decimális-bináris konverzió algoritmusa szerint). Egyszerűbb és gyorsabb módszer a tízes rendszerből kettes számrendszerbe alakítás után 4 bites csoportosítással a kettes számrendszerből a hexadecimális számrendszerbe alakítani (→bináris-hexadecimális konverzió).

decimális-oktális konverzió (decimal-to-octal conversion) – tízes számrendszerben ábrázolt számok átalakítása →oktális számrendszerbe. Algoritmusa megegyezik a →decimális-hexadecimális konverzió algoritmusával, ha a 16-ot 8-cal helyettesítik. Az egyszerűbb módszer itt is a →decimális-bináris konverzió után nyert eredmény 3 bites csoportosítása és ezek helyettesítése a megfelelő oktális jegyekkel (→bináris-oktális konverzió).

decimális számábrázolás (decimal representation) – számoknak köznapi szemléletünknek megfelelő, a 10 hatványai szerint rendezett ábrázolása, amely az

$$a_n \cdot 10^n + a_{n-1} \cdot 10^{n-1} + \ldots + a_2 \cdot 10^2 + a_1 \cdot 10^1 + a_0 \cdot 10^0$$

felel meg. Az $a_n, a_{n-1}, \ldots, a_2, a_1, a_0$ együtthatók jelkészlete 0,1,2,3,4,5,6,7,8,9. Minthogy 3 bit csak $2^3 = 8$ különböző állapotot vehet fel, egy decimális számjegy egyértelműen csak 4 bittel ábrázolható. Az, hogy a lehetséges $2^4 = 16$ kombináció közül melyik 10-et használják és azokat milyen módon feleltetik meg a decimális jegyeknek, az alkalmazott →kódtól függ. ~nál a számot mindig előjelével és abszolút értékével adják meg.

dedikált erőforrás (dedicated resource) – olyan →hardver v. →szoftver →erőforrás, amely egyetlen felhasználó számára van kiosztva mindaddig, amíg annak szüksége van rá. Jellegzetes ~ok a →rögzített partíciókkal dolgozó →operációs rendszereknél az egyes partíciókhoz rendelt →periferikus művek (pl. a →háttérpartícióhoz kiosztott standard →bemeneti egység, ill. →kimeneti egység). ~ az olyan rendszerprogram (→fordítóprogram, →kapcsolóprogram stb.) is, amely adott partícióban egy felhasználó igényeit szolgálja ki (szemben az →osztott erőforrással).

definíció (definition) – programozástechnikai szempontból olyan →deklaratív utasítás,

amellyel – az adott programozási nyelv →szintaxisa alapján – meghatározzák a →változók attribútumait. Ezekből kiindulva a →fordítóprogram kijelöli a szükséges tárolóhelyet, kiválasztja az →[adat]típusnak megfelelő →gépi utasításokat stb. A klasszikus programozási nyelvekben a ~ és a →deklaráció fogalmak fedik egymást, egyes modern programozási nyelvekben viszont különbséget tesznek közöttük.

deklaráció (declaration) – olyan →deklaratív utasítás, amellyel a programozási objektumok attribútumait meghatározzák. A programozási nyelvek nagy részénél a ~ egybeesik a →definícióval, ami azt jelenti, hogy a jellemzők leírása és az azok alapján kijelölt memóriaterület kiutalása egyszerre történik. A modern nyelvekben (→C, →Ada stb.) szétválasztották e két fogalmat. Definícióról beszélnek azon a helyen, ahol a változót ténylegesen előállítják, vagyis tárolóhelyet utalnak ki számára, ~ról viszont ott ahol a változónak csak a természetét határozzák meg, de tárolóhelyet máshol rendelnek hozzá.

deklaratív nyelv (declarative language) – a →programozási nyelvek egyik osztálya, feladatoknak nem algoritmikus megoldására. Abban különbözik az →eljárásra orientált nyelvektől, hogy elsősorban az elfogadható eredmények s nem az eredményhez vezető eljárás(ok) leírását támogatja. Elsősorban a →mesterséges intelligencia területén használatos (→logikai programozás, →LISP, →Prolog).

deklaratív utasítás (declarative statement) – leíró jellegű, →nemvégrehajtható utasítás, amely a programban felhasznált objektumok attribútumait rögzíti. A konkrét programozási nyelvtől függően ezek az attribútumok vonatkozhatnak az adatok típusára (→[adat]típus), szerkezetére, ábrázolási módjára, a memóriakiutalás módjára (statikus v. dinamikus), a belső ábrázolás pontosságára stb. ~ban rögzített előírások alapján végez el a →fordítóprogram olyan fontos tevékenységeket, mint a megfelelő tárkiutalás, a programkörnyezet és az [adat]típusok közötti összefüggések ellenőrzése, a →gépi utasítások kiválasztása az →utasításkészletből, egyes gépi utasítások hiányában a szükséges szimuláló →szubrutinok beépítése a →tárgyprogramba, az előírt műveletek elvégzéséhez szükséges →adatkonverziók beépítése ill. letiltása, az adatstruktúrák elemeinek eléréséhez szükséges címszámítások elvégzése stb.

dekódolás (decoding) – a →kódolás ellentett művelete: a kódolt információk eredeti állapotára való visszaalakítása.

dekódoló (decoder) – valamely jelsorozat értelmét megfejtő és ennek alapján a megfelelő hatást kiváltó egység.

destruktív memória (destructive memory) – olyan memóriatípus, amelynél kiolvasáskor a tárolt információ törlődik, s ezért olvasása után az információt újra be kell írni. Ilyen pl. a →ferritmemória.

determinisztikus automata (deterministic automaton) – olyan →automata, amelynek következő állapota egyértelműen meghatározott, ha ismert a pillanatnyi állapota és a bemenő szimbólum.

determinisztikus játék (deterministic game) – olyan →játék, amelyben a játékosok által választott stratégiák a játék kimenetelét egyértelműen meghatározzák.

determinisztikus program (deterministic program) – olyan →program, amelynek a végrehajtás alatti, időben bekövetkező állapotváltozásai előre meghatározhatók, vagyis a bemenő adatok ismeretében a program által szolgáltatott eredmények előre kiszámíthatók.

diagnosztizáló program (diagnostic program) – számítógépes rendszerekben fellépő gépi hibák helyének és okának felderítésére szolgáló program, a karbantartási tevékenység alapvető eszköze (pl. segítségével lokalizálhatók azok a memóriarekeszek, ahol a beírás v. kiolvasás hibásan működik). A ~ rendszerint autonóm üzemmódban, az →operációs rendszeren kívül működik, egyes modern változatai viszont lehetővé teszik, hogy egy-egy meghibásodott egységet (pl. →perifériát, memóriamodult stb.) közvetlenül az operációs rendszer felügyelete alatt vizsgáljanak anélkül, hogy az a rendszer többi részét akadályozná a munkában.

digitális adatátvitel (digital transmission) – hagyományos távíró- és telexhálózatokban használt adatátviteli módszer, amelynél a digitális információt frekvenciamódosítás nélkül

adják az átviteli vonalra. Kevésbé gyors adatátvitelre megfelel, nagyobb sebességeknél azonban →analóg adatátvitelt kell használni.

digitális-analóg konverter (digital-to-analogue converter) – adott digitális jelsorozatot olyan folytonos (analóg) jellé alakító berendezés, amelynek mérőszáma (amplitúdója) éppen a bemenő jelsorozat. Jellegzetes felhasználási területe a folytonos folyamatok irányítása →digitális számítógépekkel.

digitális jel (digital signal) – olyan jel, amely az általa jellemzett mennyiség mérőszámának megfelelően csak véges számú diszkrét, egymástól élesen elhatárolt értéket vehet fel. A számítógép-technikában szinte kizárólag csak bináris ~eket használnak (ezek csak két diszkrét értékkel rendelkezhetnek).

digitális nyomtató (digital printer) – általában minden olyan →nyomtató, amelyet a számítógépekben előállított információk digitális ábrázolású kinyomtatására használnak. Szűkebb értelemben olyan speciális készülék, amely analóg mérési eredmények digitális megjelenítésére, nyomtatására alkalmas.

digitális számítógép (digital computer) – numerikus →kódokkal ábrázolt információk feldolgozására alkalmas számítógép. A feldolgozás véges számú, diszkrét utasítás végrehajtása révén megy végbe, vagyis olyan műveletek láncolata, amelyeknél meghatározott azok típusa és a kezelt operandusok →címe. →Utasításkészletéhez alapvetően aritmetikai, logikai és adatmozgató utasítások tartoznak. A ~ legfontosabb jellemzői:

1) →Operatív memóriával rendelkezik, amely az adatok és programok időleges tárolására alkalmas;

2) Működését a tárolt program vezérli a →központi vezérlőegység révén; az utasítások végrehajtását →aritmetikai és logikai egység végzi;

3) A programok és az adatok egy v. több →bemeneti egységen keresztül juttathatók az operatív memóriába;

4) A feldolgozás eredményei egy v. több →kimeneti egységen keresztül jutnak el a felhasználókhoz;

5) Az összetevő funkcionális egységek koordinált vezérlését a központi vezérlőegység, ill. az operációs rendszer látja el.

digitalizálás (digitization) – analóg adatok átalakítása digitális (numerikus) adatokká. Különleges jelentőségű a grafikus információk ~a, ennek két alapvető formája ismeretes.

A koordináta-~ főként vonalas ábrák (műszaki rajzok, térképek, kapcsolási rajzok stb.) feldolgozására alkalmas. Digitális információként az ábra bizonyos pontjainak koordinátáit használják.

A raszteres ~ elsősorban tónusos ábrák (fénykép- v. röntgenfelvételek stb.) feldolgozására alkalmas. A kép egészét elemi pontokra bontják, majd e pontok pozícióját, tónusát, esetleg színösszetételét alakítják át numerikus értékekké.

A grafikus információ ~a külön problémát is felvet: a digitális információelemek önmagukban még nem fejezik ki a képi információ jelentéstartalmát. Rendkívül bonyolult programok szükségesek ahhoz, hogy a képi információelemek digitális átalakítása mellett együttes jelentésüket is kódolni lehessen.

dinamikus adatstruktúra (dynamic data structure) – olyan →adatstruktúra, amelynek szerkezeti felépítése időben változó (pl. →fa, →lista stb.). →Operatív memóriabeli ábrázolásához egy sor járulékos információ szükséges, a struktúrához tartozó elemek egymás közti viszonyának kifejezésére (pl. →láncolás).

dinamikus áthelyezés (dynamic relocation) – programok áthelyezésének az a módja, amikor a →logikai címtartomány leképzése az →operatív címtartomány egy részére a programok végrehajtása közben történik. Megvalósításának különböző módozatai lehetnek. →Bázisregiszterek használatával pl. egy programrészlet egyszerűen áthelyezhető, ha a hozzá tartozó bázisregiszter tartalmát megfelelően változtatják. →Átfedési szerkezet használatával az egymástól független, időben eltolódó végrehajtású programrészeknek azonos memóriazónákat utalnak ki, és a pillanatnyi körülményeknek megfelelően töltik be őket. →Lapkezeléssel a végrehajtandó programnak csak az adott pillanatban szükséges →lapjait (→munkahalmazát) töltik az operatív memóriába, mindig az éppen szabad v. felszabadított zónákba.

dinamikus attribútum (dynamic attribute) – valamely szintaktikai elem tulajdonsága,

amely csak a →programok végrehajtása idején konkretizálódik, tehát a →fordítóprogram az általa érintett →szemantikai szabályokat nem képes ellenőrizni.

dinamikus memóriakiosztás (dynamic memory allocation) – 1. Az →operatív memóriának mint →erőforrásnak olyan kiutalása, amely mindig a pillanatnyi körülményeknek megfelelően történik. Legismertebb formái: a változó hosszúságú →partíciók alkalmazása, a →lapkezelés, a →virtuális memóriacímzés implementálása. – 2. Egyes programozási nyelveknél (pl. →PL/1, →Pascal, →Ada stb.) az a lehetőség, hogy a programozó a program végrehajtása közben dönthet memóriazónák lefoglalásáról, ill. felszabadításáról (→dinamikus változó).

dinamikus modell (dynamic model) – időben lefolyó jelenségek vizsgálatára használt matematikai modell. Lényege az, hogy figyelembe veszi a változók időtől függő természetét (szemben a →statikus modellel).

dinamikus multiplexelés (dynamic multiplexing) – az →időosztásos multiplexelés olyan formája, amelynél az időkiosztás az →adatcsatornák igényeinek megfelelően valósul meg.

dinamikus szubrutin (dynamic subroutine) – olyan ismételten végrehajtott programegység, amelynél minden aktivizálás előtt →paraméterek megfelelő beállításával választják ki az igényelt funkció(ka)t.

dinamikus változó (dynamic variable) – egyes programozási nyelvekben (pl. →PL/1, →Pascal, →Ada stb.) használható olyan →változó, amelynél a tárterület kiosztása, ill. a hozzá rendelt memória felszabadítása a programozó utasítása alapján történik tetszőleges időpontokban. Elsősorban olyan adatstruktúrákra vonatkozik, amelyeknek tényleges tárigénye csak a program végrehajtásának idejében válik ismertté. A dinamikus tárfoglalás igen hasznos eszköz, mert segítségével a programozó a program igényeitől teheti függővé a pillanatnyi tárfelhasználást.

direkt alap-adatelérési technika (basic direct access method) – a →direkt szervezésű [adat]állományok feldolgozására alkalmazott →alap-adatelérési technika.

direktíva (directive) – az →asszembler (ill. a →fordítóprogram) működését vezérlő →utasítás, amely a →fordítás menetére, bizonyos adminisztratív feladatok végrehajtására vonatkozik, és nem érvényesül közvetlenül a lefordított →tárgyprogramban. Általános jelleggel használatos ~k vonatkoznak a memóriazónák lefoglalására és szerkezeti leírására, egyes →azonosítók globális jellegének deklarálására, másoknak mint →külső hivatkozásoknak a definiálására stb. Ugyancsak ~ segítségével lehet előírni a →forrásprogram listázásának elnyomását, a listázás formájára és tartalmára vonatkozó elvárásokat, egyes különleges szolgáltatások igénylését (pl. →kereszthivatkozások listájának összeállítását) stb.

direkt kód (direct code) – számoknak a számítógépben alkalmazott olyan bináris ábrázolása, amelyben a rendelkezésre álló n bitpozíció közül n–1 pozíción a szám abszolút értékét tárolják, az →előjelbit pedig 0 a pozitív, 1 a negatív számoknál. ~ használatával a multiplikatív műveletek azonnal elvégezhetők (másik gyakori forma a →komplemens kód).

direkt lista (push-up list) – speciális szerkezetű →lista, amelybe új adatelem rögzítése csak a mindenkori utolsó adatelem után lehetséges, kiolvasni, ill. törölni pedig csak a mindenkori legrégebben rögzített adatelemet lehet. Tkp. a →várakozási sorhoz hasonló adatszerkezet.

direkt szervezésű [adat]állomány (direct file) – nevezik közvetlen hozzáférésű, ill. szelektív [adat]állománynak is. Olyan →logikai [adat]állomány, amelynél a →logikai rekordok között semmifajta logikai sorrendet nem lehet megállapítani, viszont közvetlen kapcsolat teremthető egy, a rekordot jellemző tulajdonság és a rekord által a címezhető →adathordozón elfoglalt helye között. Mivel a rekordok feldolgozási sorrendje esetenként változik, ezt a szervezési módszert „véletlen hozzáférésű" szervezésnek is nevezik. A logikai rekordok szimbolikus kulcsai és a fizikai címek között leképzési algoritmusok teremtik meg a kapcsolatot (→terítőfüggvény). A ~ legkisebb címezhető egysége a kazetta, amelybe a szervező által meghatározott számú rekord fér el. A különböző kulcsokkal, de azonos fizikai címmel rendelkező rekordokat szinonimoknak nevezik. Az [adat]állomány előállításakor mindazok a rekordok, amelyek

elosztási algoritmusuk során azonos kazettaszámot kapnak (tehát szinonimok), szekvenciálisan helyezkednek el az illető kazettában, majd annak megtelte után láncolt formában a túlcsordulási területre kerülnek. Adott rekord elérése így a megfelelő kazetta direkt címzése, majd a kazettán belül és esetleg a túlcsordulási területen szekvenciális keresés útján valósul meg.

diszjunkció (disjunction) – →logikai összeadás

diszjunktív normálalak (disjunctive normal form) – adott →logikai függvény argumentumában szereplő változók és negáltjaik konjunkcióinak diszjunkciója (logikai szorzatainak logikai összege), amiből minden összetett ítélet előállítható. Ha az f(x,y) logikai függvényt a következő igazságtáblázat definiálja:

x	y	f
0	0	0
0	1	1
1	0	1
1	1	0

akkor a hozzá tartozó ~: $E(x,y) = x \cdot \bar{y} + \bar{x} \cdot y$

diszkrét változó (discrete variable) – olyan →változó, amelynek értékei megszámlálható halmaz elemei közül kerülhetnek ki.

dokumentummásoló (hard copy) – →párbeszédes üzemmódban dolgozó rendszerekben az a berendezés, amely későbbi felhasználás céljából papírra rögzíti az →interaktív terminál képernyőjén átfutó információkat (az angol terminológia egyben a berendezésnek a termékét is jelöli).

DOS (Disk Operation System) – az IBM System-360 számítógépcsalád egyik mágneslemezes →operációs rendszere, amely hosszú ideig meghatározó volt a hasonló programok megvalósításában. A ~ mozaikszót általában is használják a mágneslemez-orientált operációs rendszerekkel kapcsolatban (pl. →MS-DOS).

döntési fa (decision tree) – döntési tevékenységeket használó →algoritmusok, tevékenységek, feltételes műveletek stb. ábrázolására használt →fa. Formálisan a csomópontok egy-egy döntést, csúcsai pedig a döntéssorozat eredményeit képviselik. A vezérlés rendszerint a gyökérelemből indul, és a döntéseknek megfelelő részfákon haladva jut el az eredményekhez.

döntési szabály (decision rule) – a →döntési táblázat jegyzékterületének egyik oszlopa, amelyhez a feltételek állapotvektora és a végrehajtandó tevékenységvektor tartozik. Szavakkal kifejezett értelmezése: ha fennáll a lehetséges feltételek adott állapotkombinációja, akkor elvégzendő(k) a következő tevékenység(ek). A döntési táblázatok teljessége szempontjából különleges jelentősége van az →„egyébként" szabálynak. Ez annak a különleges esetnek felel meg, amikor a →feltételjegyzék egyik esete sem tartalmazza a pillanatnyi állapotot.

A ~oknak két változata használatos.

Az ÉS típusú ~:

> ha feltétel-1 és feltétel-2 és ... igaz, akkor elvégzendő(k) a következő tevékenység(ek).

A VAGY típusú ~:

> ha feltétel-1 vagy feltétel-2 vagy ... igaz, akkor elvégzendő(k) a következő tevékenység(ek).

Az ÉS típusú döntési táblázatok az elterjedtebbek, és ha nincs külön utalás a táblázat fajtájára, akkor azt egyértelműen ÉS táblázatnak tekintik.

döntési táblázat (decision table) – rendszerek, algoritmusok feltételeinek és a hozzájuk tartozó tevékenységeknek szabványos, tömör ábrázolására használt módszer, ill. maga az ilyen eljárással készült táblázat. A ~ téglalap alakú dokumentum formájában tartalmazza az adott feladatra vonatkozó összes információ leírását a matematikai logika eszközeinek segítségével.

DS (Double Side) – kétoldalas →hajlékonylemez típusjele.

duálrendszer (dual system) – két számítógép összekapcsolása révén nyert rendszer. Célja az, hogy ugyanazoknak a programoknak a párhuzamos végrehajtásával a feldolgozás hibamentességének biztonságát növeljék (ikerrendszer), v. hogy két kisebb gépből nagyobb rendszert létrehozva az új számítógéprendszer kapacitását, hatékonyságát növeljék.

dupla pontosság (double precision) – 1. Számítógépeknél az a lehetőség, hogy

dupla hosszúságú szavakkal is végezhetők műveletek. Ilyenkor az operandusok két „összekapcsolt" rekeszben helyezkednek el. – 2. →Aritmetikai egység működésében a nagyobb pontosság elérése érdekében használt módszer, amelynél a normális (egy szó) hosszúságú adatokkal műveletet (pl. szorzást) végezve dupla hosszúságú →regisztereket használnak. Így a részműveleteknél a kerekítési hibák nem befolyásolják érzékenyen a művelet végeredményét.

duplex csatorna (duplex channel) – adatátvitelben használatos →kommunikációs csatorna, amely mindkét irányban egyidejűleg viheti át a hasznos adatokat és a vezérlőjeleket. Kis sebességeknél megoldható az átvitel egyetlen vezetékpárral is, →frekvenciaosztás segítségével. Közép- és nagy sebességeknél rendszerint négyhuzalos összeköttetést használnak.

duplex összeköttetés (duplex communication) – az →adathálózat két →adatállomása közötti olyan kapcsolat, amelyben mindkettő egyidejűleg adja és veszi a hasznos információkat, az összeköttetés mindkét irányában egyidejűleg fennáll az információáramlás. A ~hez jóval bonyolultabb adatállomási berendezések tartoznak, mint a →szimplex összeköttetéshez, ezért főleg a magas szintű hálózatágakban használják.

E, É

E – a →hexadecimális ábrázolás jelkészletének eleme, amely a decimális 14 értéknek felel meg (bináris megfelelője: 1110).

EAN (European Article Numbering) – európai cikkszámozás →vonalkódon alapuló azonosító →kódja. 8 és 13 pozíciós formája használatos. Az utóbbinál az első két (decimális) számjegy az ország, a következő 5 számjegy az előállító vállalat, az azt követő 5 számjegy a termék azonosítására szolgál, az utolsó pedig az →ellenőrző számjegy.

EBCDIC-kód (EBCDIC-code) – igen elterjedten használt 8 bites →kód az angol ábécé 26 betűjének, a 10 decimális számjegynek, az írásjeleknek, a műveleti jeleknek, valamint az adatátvitel vezérlőkaraktereinek kódolására. 256 kódszavas kód.

effektív cím (effective address) – az a tényleges →operatív memóriacím, amelyet a →központi vezérlőegység a műveletben részt vevő operandus elérésére közvetlenül felhasználhat. Az ~ lehet →abszolút cím, de legtöbbször →címszámítás eredményeként jön létre.

EGA (Enhanced Graphics Adapter) – →szöveges módban és →grafikus módban egyaránt működő, magas →felbontást kezelő →képernyőadapter. Legismertebb üzemmódja a 640x350 →képpont felbontású, amelynél a képernyőn egyszerre 16 szín lehet.

egér (mouse) – →személyi számítógépekhez kapcsolódó, a →billentyűzet →nyílbillentyűinek megfelelő funkciót ellátó vezérlőeszköz. Segítségével a →helyőr képernyőn elfoglalt helyét lehet változtatni.

egész [adat]típus (integer [data] type) – majdnem minden magas szintű programozási nyelv (→FORTRAN, →PL/1, →COBOL, →Pascal stb.) által definiált [adat]típus. Értékei az egész számok halmazából kerülnek ki, számosságuknak csak a számítógépek véges kapacitása szab határt. Minden implementációban van egy olyan számkorlát, amelynél nagyobb abszolút értékű szám az adott gépen nem ábrázolható. Pl. a 32 bites gépi →szóval rendelkező számítógépeknél ez az érték $2^{31}-1$. Az ~sal aritmetikai, összehasonlítási és értékadó műveletek végezhetők.

egész konstans (integer constant) – tizedespont nélküli →numerikus konstans.

egzisztenciális kvantor (existential quantifier) – →predikátum

egycímű számítógép (one-address computer) – olyan számítógép, amelynek utasításaiban a műveleti kódon kívül egyetlen →cím szerepel. Ez a megoldás rendszerint az olyan számítógépekre jellemző, amelyeknek →aritmetikai egysége egyetlen →akkumulátort tartalmaz. A kétargumentumú műveletek egyik argumentumát a művelet megkezdése előtt az akkumulátorba kell tölteni, az eredmény pedig szintén itt tárolódik.

„egyébként" szabály („else" rule) – a →döntési táblázatoknál a →döntési szabály speciális esete, amely abban a különleges helyzetben lép érvénybe, amikor a →feltételjegyzék egyik bejegyzése sem tartalmazza a pillanatnyi állapotot. Más szóval, ha az egyenként felsorolt szabályokban a feltételek kombinációinak egyike sem érvényes, akkor az ~ban megadott tevékenység(ek)et kell végrehajtani. Ennek az esetnek különleges jelentősége lehet a számítógépes felhasználáskor: ha a feltételek egy bizonyos kombinációját a program tervezésekor elmulasztották figyelembe venni, az ~ba pedig hibajelzéses megállást iktattak, a tervezés (v. megvalósítás) egyes logikai következetlenségei még idejében felfedezhetők és kijavíthatók.

egyenletes összeválogatás (balanced merging) – az →összeválogatásnak a külső →rendezés megvalósítására használt módszere. Négy külső →periferikus művet (A, B, C, D) feltételezve, az ~ menete a következő. A rendezendő eredeti →[adat]állományból, amennyit lehet, beolvasnak az →operatív memóriába, s ott valamilyen →belső rendezési módszerrel előállítják az első monotóniát, majd kiírják az A perifériára (A1). A hasonló módon előállított következő monotóniát a B perifériára írják ki (B1), majd váltakozva a többit (A2, B2, A3, B3 stb.). Ezután következik a tulajdonképpeni ~. Rendre beolvassák az A1 és B1 elemeit az operatív memóriába, és a rendezési kulcsnak megfelelően egyenként, az előírt sorrendben előállítanak egy újabb monotóniát a C periférián. Hasonló módon állítanak elő újabb monotóniákat az A2 és B2 elemeiből, és így tovább, amíg az A és B pe-

rifériák ki nem ürülnek. A következő lépésben a C-ről és D-ről válogatják össze páronként a rendezett részhalmazok elemeit, és az A és B periférián gyűjtik össze az újabb monotóniákat. Az eljárást addig folytatják, amíg végül egyetlen rendezett halmazt kapnak. A műveleteket természetesen több periferikus műre is ki lehet terjeszteni, lényegesen csökkentve a teljes rendezési időt.

egyirányú lista (one-way list) – olyan →lista, amelyben minden adatelemnek egyetlen követője van, vagyis az adatstruktúra bármelyik eleme csak az első adatelemből kiindulva érhető el.

egymásbaágyazás (nesting) – a programozástechnikának az a módszere, amely iterációs →ciklusok hierarchikus egymás fölé rendelésében nyilvánul meg. Pl. a →FORTRAN, az →ALGOL, a →PL/1, a →Pascal stb. megenged további iterációs utasításokat a hasonló típusú utasítások keretein belül is. Az iterációknak ez az ~a sok számítási művelet elvégzésekor igen lényeges, és igen jelentős eszköz a →strukturált programozás módszertanában is. Két egymásba ágyazott ciklus nem kerülhet fedésbe, ami azt jelenti, hogy az egyik ciklus összes utasításának hozzá kell tartoznia a másik ciklus törzséhez. Ugyanez az alapelv érvényesül a →blokkstruktúrákat használó programozási nyelveknél az →eljárások szintjén.

egyszerű feltétel (simple condition) – →programozási nyelvekben használatos →szintaktikai kategória, amely kétalternatívás programelágaztatást tesz lehetővé. A →COBOL-ban pl. a következő ~ek ismertek: →típusfeltétel, →feltételnév-feltétel, →előjelfeltétel és relációfeltétel.

egyszerű név (simple name) – →programozási nyelvekben használható olyan →azonosító, amelyre önmagában egyértelműen lehet hivatkozni, mert más azonos szimbólum nem szerepel a programban (szemben a →minősített névvel).

egyszintű memória (one-level memory) – olyan →memóriahierarchia, amely a felhasználó szempontjából egységes, azonos hozzáférési lehetőséget biztosító, gyakorlatilag bármilyen kapacitású memória látszatát kelti. Tkp. a →virtuális memóriához nagyon közel álló fogalom. Lényege az, hogy a →központi memória és a →háttértár közötti →adatvándorlást a →hardver, ill. a →szoftver speciális eszközeinek segítségével a felhasználó számára transzparens módon oldja meg.

ekvivalencia (equivalence) – kétargumentumú →logikai függvény, amely akkor és csakis akkor igaz, ha a két argumentum értéke azonos. Jelölése A≡B, igazságtáblázata pedig

≡	0	1
0	1	0
1	0	1

elágazás (branch) – →algoritmusokban v. az azokat megvalósító →programokban előforduló döntési pont; valamilyen feltételt kiértékelő vizsgálat eredménye dönti el a következőkben végrehajtandó lépéseket.

elárasztásos adatútképzés (flood routing) – olyan →adatútképzés, amelynél a forráscsomópont mindegyik szomszédos csomóponthoz elküldi az →adatcsomagot. Minden egyes csomópont, amely adatcsomagot fogad, megvizsgálja, hogy megelőzően azt fogadta-e már, és ha igen, figyelmen kívül hagyja. Amikor első alkalommal fogad egy adatcsomagot, az összes szomszédjához továbbítja, kivéve azt, ahonnan az illető adatcsomag érkezett. Az eljárás előnye a csomópont- és összeköttetés-hibákkal szembeni rugalmasság, viszont jelentős többletforgalmat eredményez.

elektromosan törölhető permanens memória (electrically erasable read-only memory) – olyan →permanens memória, amelynél az összes bit egyidejű törlését el lehet végezni elektromos impulzus segítségével.

elektronikus posta (electronic mail) – egymással →adatátviteli hálózaton keresztül kapcsolatban álló →munkaállomások azon lehetősége, hogy egymással információkat cserélhessenek. Ennek érdekében a rendszer minden résztvevője rendelkezik egy „postaládával", ahol a feladó tárolja, s ahonnan a címzett lehívhatja a neki szóló üzeneteket. Újabban az ~t →városi hálózatok is lehetővé teszik mint nyilvános szolgáltatást.

elektrosztatikus memória (electrostatic memory) – az első generációs számítógépek egy részénél használt memóriatípus. Megoldása technikailag a televíziós képcsövekéhez

hasonlít, azzal az eltéréssel, hogy az anód a képcső ernyőjén helyezkedik el, előtte pedig dielektrikumlemez van. Ebben tárolódik az információ az anódból kilépő szekunder elektronok formájában.

elektrosztatikus nyomtató (electrostatic printer) – olyan →nyomtató, amelynél a nyomtatóblokk mátrixszerűen elhelyezett elektródokból áll. A nyomtatandó karakter rajzának megfelelően szelektíven kisütött elektródok feltöltik a speciális papírt. A papírt festékporon vezetik keresztül, ezalatt a statikusan feltöltött felületek magukhoz vonzzák a festékport, amelyet később beolvasztanak a papírfelületbe. A mozgó részek hiánya igen nagy nyomtatási sebességet biztosít, hátránya viszont, hogy nem ad lehetőséget másolatok közvetlen előállítására.

elérés (access) – →hozzáférés

elérési idő (access time) – az az időtartam, amely a →cím rendelkezésre állásától az adott címen tárolt információegység (→bájt, →szó stb.) eléréséig szükséges. Nagyságrendje és összetevői különbözőek annak függvényében, hogy az →operatív memóriával v. valamely →háttértárral kapcsolatban vizsgálják.

elhatárolójel (delimiter) – →programozási nyelvek →szintaktikai kategóriája: olyan jel, amely a szintaktikai kategóriák elválasztására alkalmas. Leggyakrabban a szóköz karaktert használják erre a célra, de ilyen szerepet játszhatnak a műveleti jeleket szimbolizáló speciális karakterek, a pont, a vessző, a pontosvessző, a zárójelek, a szóköz stb. is. Az ~nek fontos szerepe van a →lexikális elemzésben.

eljárás (procedure) – 1. Valamely probléma megoldása céljából végrehajtott (véges számú) műveletek sorozata. – 2. Olyan önálló program v. programrészlet, amely más programokba beépíthető és egyetlen hívóutasítással aktivizálható (kb. ugyanaz a fogalom a magas szintű programozási nyelvek szintjén, mint a →szubrutin az →asszembler nyelveknél). Azonosítására az eljárásdeklarációt használják, amely rögzíti az ~ azonosítóját (ez szerepel majd a hívóutasításban), a →formális paraméterek számát, típusát, sorrendjét, valamint a paraméterátadás módját (→érték szerinti hivatkozás, →cím szerinti hivatkozás, →név szerinti hivatkozás) v. ezek kombinációját.

Az →aktuális paraméterek hozzárendelése a formális paraméterekhez a hívás pillanatában történik. – 3. Az →adatátviteli hálózatokban az információcsere lefolyására és módjára vonatkozó szabályok és előírások összessége.

eljárásdeklaráció (procedure declaration) – valamennyi eljárásra orientált, magas szintű programozási nyelvre jellemző →deklaratív utasítás, amely rögzíti az eljárás azonosítóját, típusát és az eljárás algoritmusának leírására használatos →formális paramétereket (szám, típus, sorrend, paraméterátadási mód stb.).

eljáráshívás (procedure call) – →eljárás aktivizálására elvégzett tevékenységek összessége. Elvileg a következő fontosabb feladatokat kell megoldania: a hívott eljárás →formális paramétereinek megfelelő, a konkrét végrehajtásához szükséges →aktuális paraméterek előállítása, a vezérlés átadása a hívott eljárásnak, az esetleges eredmények átvétele, a vezérlés visszanyerése. A legfontosabb művelet, amely az ~ típusát is meghatározza, az aktuális paraméterek átadása. Ennek függvényében az ~ történhet →cím szerinti hivatkozással, →érték szerinti hivatkozással v. →név szerinti hivatkozással.

eljáráshívó utasítás (procedure call statement) – magas szintű →vezérlésátadó utasítás, amelynek segítségével a →moduláris programozás elvei alapján írt programok aktív moduljai bizonyos logikai sorrend szerint aktivizálhatók. Az ~ tartalmazza a hívott →eljárás nevét (azonosítóját) és a →formális paramétereknek megfelelő →aktuális paramétereket. A hívott eljárás végrehajtása után a vezérlést automatikusan az ~t szekvenciálisan követő utasítás kapja.

eljárási főrész (procedure division) – →COBOL programozási nyelven írott programok operatív része, vagyis az a rész, amelyben a végrehajtandó algoritmust írják le.

eljárásmodul (procedure module) – a →moduláris programozás elvei alapján előállított program olyan aktív →modulja, amelynek feladata egy v. több jól meghatározott tevékenység, feldolgozás megvalósítása. Az ~okat a program sorrendi logikájának meg-

felelően a →vezérlőmodul aktivizálja, de hívhatók egymás között is.

eljárásra orientált nyelv (procedure-oriented language) – magas szintű →programozási nyelv, amely lehetővé teszi az →algoritmusok számítógépes megfogalmazásában a klasszikus megoldáshoz igazodó formalizmus használatát. A →deklaratív nyelvektől eltérően elsősorban az eredményekhez vezető eljárás(ok) leírását támogató eszközökkel rendelkezik. Jellegzetes –ek: →COBOL, →PL/1 (gazdasági-ügyviteli adatfeldolgozásra), →ALGOL, →FORTRAN, →PL/1 (műszaki-tudományos célokra), →B, →BCPL, →C, →CDL, →Pascal, →Ada, →CHILL (rendszerprogramozásra) stb. Alapvető jellegzetessége, hogy a programozó az eljárás közben fellépő állapotoknak megfelelően maga vezérelheti programját: kezdeményezi az adatmozgatást, vezérli a ciklusokat, összetett feltételtevékenység utasításkomplexumokat építhet fel stb.

ellenőrzőbit (control bit) – 1. →Aszinkron átvitelnél a hasznos információt hordozó karaktert határoló →bit. Minden egyes karaktert egy –, a START-bit előz meg, és egy v. több –, STOP-bit zár le. Az adó és vevő adatállomások csupán az –ek közötti intervallumokban vannak egymáshoz szinkronizálva. – 2. Átviteli és tárolási hibák felismerésére szolgáló járulékos bit. Minden egyes →bájthoz –et rendelve, a benne szereplő 1-es bitek számát jellemezni lehet. Az –et úgy állítják 1-es v. 0-s értékre, hogy a bájtban található összes 1-et tartalmazó bitek összege a rendszerre jellemzően páros, ill. páratlan értéket adjon (paritásbit). Adatátvitelkor az adó gondoskodik az – hozzáadásáról, míg a vevőoldalon ellenőrizhető a beérkezett 1-es bitek száma. Eltérésnél az átvitel hibás. – használatával az egybitnyi adattorzulások észlelhetők.

ellenőrzőpont (checkpoint) – 1. Programokban előre megállapított hely, ahol méréseket, kiértékeléseket, ellenőrzéseket végezhetnek az illető programok tesztelése, javítása v. a hatásfok ellenőrzése céljából. – 2. A programhelyesség automatikus bizonyítása érdekében bevezetett állítások, ill. azok kiértékelési pontja.

ellenőrző számjegy (control digit) – a →numerikus adatok →redundanciáját növelő számjegy, amelyet az illető adatok helyességének ellenőrzésére használnak. Meghatározása az adatok kódolásánál, ellenőrzése az adatok felhasználásánál történik. Mindkét művelet lényege általában az, hogy az illető adat pozícióihoz bizonyos helyértéket kapcsolnak, majd polinomműveleteket végeznek velük valamilyen modulo aritmetikának a szabályai szerint.

ellenőrzött végrehajtás (controlled execution) – programok belövési (tesztelési) fázisában a kódolási, ill. logikai hibák felderítésére használt módszer, amelynek célja a nagy sebességű végrehajtás lelassítása az emberi kiértékelhetőség szintjére. Lényege hasonló a →lépésenkénti végrehajtás módszeréhez, azzal a különbséggel, hogy az – nagyobb programrészekhez tartozó funkciók vizsgálatára ad lehetőséget. Míg az előzőnél a működés soha nem lehet azonos idejű, az –nál már a programszakaszon belül azonos idejű működés érhető el.

Elsősorban a →párbeszédes üzemmódban dolgozó rendszereknél terjedt el, és a következő lehetőségeket teremti meg a programozó számára: 1) a program végrehajtásának előre meghatározott pontokon történő felfüggesztése és esetleges változtatások utáni újraindítása v. folytatása; 2) a program állapotát meghatározó adatok (memóriarekeszek, regiszterek) kiírása s ezáltal a program helyes működésének a részeredmények alapján történő ellenőrzése; 3) memóriarekeszek, regiszterek tartalmának felülírása; 4) egyes bejárt programágak vizualizálása stb.

elosztott adatbázis (distributed data base) – az →elosztott feldolgozás elveinek megfelelő →adatbázis. Lényegében kétféle módon alakítható ki.

Az egyik módszer szerint a →fizikai adatbázist tagolják részekre és tárolják az elosztott rendszer különböző helyein. Az így kialakított adatbázis particionált adatbázis, eredetileg rendszerint centralizáltként létesült, a részekre bontás csak később következett be. E módszer elsőrendű célja az, hogy az egyes adatelemeket a lehető legközelebb vigyék a felhasználóhoz. Alapvető jellemzője, hogy minden adatelem csak egyszer fordul elő az adatbázisban.

A második módszer a →logikai adatbázist tagolja és helyezi ki a rendszer legalább két pontjára. Alapvető jellemzője az, hogy egyes adatelemek több helyen is előfordulhatnak, esetleg a teljes adatbázis is, az elosztott rendszer különböző feldolgozási pontjain elhelyezett másolatok átfedhetik egymást. Rendszerint ott használják, ahol a felhasználói igények nem csoportosíthatók területileg. Bár a többszörös másolatok javítják a hozzáférhetőséget és az adatbiztonságot, a különböző másolatok egymástól független aktualizálása jelentősen lerонthatja az adatbázis hatékonyságát.

elosztott feldolgozás (distributed processing) – olyan feldolgozási mód, amelyben a feldolgozó egységek és maga a feldolgozási folyamat egymással kapcsolatban álló, de fizikailag és/vagy funkcionálisan elkülönülő részekre oszlanak. ~t valósít meg két v. több, egymással kapcsolatban álló számítógép, két v. több, területileg szétszórt feldolgozó egység, amelyek között a fizikai kapcsolat →kommunikációs csatornákon keresztül jön létre. Alapvető jellegzetessége, hogy az alkotó elemek összehangolását, közös vezérlését egyetlen, az egész rendszerre kiterjedő →operációs rendszer végzi. Az ~ és a →számítógép-hálózatok megjelenése talán a legjellemzőbb vonása a →negyedik generációs számítógépeknek.

elosztott vezérlés (distributed control) – az →elosztott feldolgozással dolgozó →operációs rendszer üzeme. A klasszikus rendszervezérlési feladatokon kívül az ilyen típusú operációs rendszernek vezérelnie kell a különböző hálózati komponensek összehangolt működését, az átviteli vonalak kiosztását, ütemezését, a távoli →adatállomások irányítását is.

előállított azonosító (created identifier) – →fordítóprogramok által a →tárgyprogram előállítására képzett belső →azonosító, amely a tárgyprogramban megjelenik, de a →forrásprogramban nincs megfelelője. A tesztelés megkönnyítésére, ill. a forrásprogram azonosítóitól való megkülönböztetésül, az ~ rendszerint olyan különleges karaktert tartalmaz, amely nincs benne az illető programozási nyelv jelkészletében. Pl. az „IF feltétel THEN akció-1 ELSE akció-2" →COBOL nyelvű struktúra tárgynyelvű átalakítása a következő programrészletnek felel meg:

feltétel kiértékelése;
ha az eredmény hamis, ugrás C%001-re;
 akció-1 kifejtése;
 ugrás C%002-re;
C%001 akció-2 kifejtése;
C%002 ...

A fenti programrészletben C%001, ill. C%002 ~k.

előfeldolgozás (preprocessing) – valamely programtermék bemenő adatain végzett előkészítő műveletek összefoglaló megnevezése (ismertebb műveletek: a szűrés, az →adat-előkészítés, az →adatérvényesítés stb.). Jellegzetes ~ pl. az →előfordítás v. a hiteles →gépi fordítás támogatására végzett kétértelműség-feloldások.

előfizető (abonent) – az →adatátviteli hálózatok szolgáltatásait igénybe vevő felhasználó. Igényesebb meghatározásban mindaz, ami a hálózat és a külvilág kapcsolatát megvalósító →interfészen kívül van. Így az adathálózatban a számítógépet és a →terminálokat összekötő →multiplexer tkp. ~.

előfordítás (precompiling) – a →makrofeldolgozás elvének kiterjesztése a magas szintű →programozási nyelvekre. Lényegében bizonyos szöveges információ lefordítását jelenti az L magas szintű nyelvre. A lefordítandó szöveg struktúrái az L specifikus struktúráival kifejezhető szerkezetek, amelyeket az ~ során bontanak le. Jelentősége, hogy már kész →fordítóprogrammal rendelkező programozási nyelvet új struktúrákkal bővíthetnek anélkül, hogy nagy komplexitású új fordítóprogramot kellene írni, ugyanis az ~t sokkal egyszerűbb program is megvalósíthatja.

Az ~ lehetőséget ad pl. arra, hogy a →vezérlési struktúrákban viszonylag szegény →FORTRAN nyelvet a →Pascalból ismert új struktúrákkal gazdagítsák:

[címke] WHILE (feltétel) utasítás

Pascal-orientált struktúra viszonylag könnyen kifejezhető a következő FORTRAN nyelvű programrészlettel:

```
[címke] CONTINUE
  1       IF (.NOT.(feltétel)) GO TO 2
```

utasítás
GO TO 1
2 CONTINUE

előindexelés (preindexing) – a →címszámítás olyan módszere, ahol az →indexregiszter tartalma az →indirekt címzés előtt veszik figyelembe (szemben az →utóindexeléssel).

előjel-ábrázolás (sign representation) – →numerikus adatok előjelének előállítása az →operatív memóriában. Az adatok ábrázolási módjának megfelelően többféle módszere ismeretes.

→Fixpontos számábrázolásnál –ra rendszerint a bitvektor bal szélső (legmagasabb helyértékű) bitjét használják:

+/– abszolút érték v. →komplemens kód

Mivel a (+/–)0 ábrázolása ugyanazt az értéket képviseli, az ábrázolás valódi modulusa eggyel kisebb, mint a lehetséges kombinációk száma.

→Lebegőpontos számábrázolásnál az –hoz tkp. két bitre volna szükség, egyre a →mantissza, egyre pedig az exponens számára. Ezt rendszerint úgy küszöbölik ki, hogy csak a mantissza előjelének tartanak fenn bitpozíciót (a bitvektor bal első bitjét), az az exponens helyett pedig a →karakterisztikát ábrázolják. Ha az exponens számára használt bitpozíciók száma 7, akkor ennek modulusa 128. A karakterisztikát úgy határozzák meg, hogy a legnagyobb pozitív érték 128 legyen: C = 64+E, ahol az E az exponens algebrai értéke. Ha a mantissza negatív, a karakterisztika az abszolút értékének komplementuma lesz. Innen következik, hogy ennél az ábrázolásnál egy negatív szám az ábrázolási bitvektor komplementuma. Ezzel a módszerrel elérhető, hogy az összehasonlítást a fixpontos műveletekkel lehessen elvégezni.

→Zónázott decimális ábrázolásnál minden →bájt egy decimális számot tartalmaz. A jobb szélső bájt kivételével mindegyik a decimális számot megelőzően az „1111" tetrádot (hexadecimális F-et) tartalmazza, a jobb szélső pedig az előjel ábrázolására választott tetrádot (a „+" számára rendszerint az A, C, E, F, ill. a „–" számára B v. D). Pl.

a +1234 ábrázolása
F1 F2 F3 F4 v. F1 F2 F3 C4

a –1234 ábrázolása
F1 F2 F3 B4 v. F1 F2 F3 D4

→Tömörített decimális ábrázolásnál minden bájt két decimális számjegyet tartalmaz, kivéve a jobb szélsőt, amely az utolsó decimális jegyet és az előjel számára választott tetrádot tartalmazza. Pl.

a +1234 ábrázolása 01 23 4C

a –1234 ábrázolása 01 23 4B

előjelbit (sign bit) – →numerikus adatok →fixpontos v. →lebegőpontos számábrázolásában az előjel kódolására felhasznált bitpozíció. Rendszerint az ábrázoló bitvektor bal első bitje (→előjel-ábrázolás).

előjelfeltétel (sign condition) – a →COBOL programozási nyelvben használatos →egyszerű feltétel, amely programszétágaztatást tesz lehetővé valamilyen numerikus operandus aktuális értékének előjele alapján (formái: POSITIVE, NEGATIVE, ZERO).

előlapozás (prepaging) – →lapkezelésnél használt helyettesítési módszer. Lényege, hogy a →lapokat az →operatív memóriába töltik, még mielőtt a programnak – végrehajtása közben – ténylegesen szüksége lenne rájuk. A módszer előnye, hogy csökkenti a →laphibák előfordulási sűrűségét, de előnytelen abból a szempontból, hogy a lapváltás mennyiségét esetleg olyan lapok betöltésével is növeli, amelyekre nem fognak ténylegesen hivatkozni a közeljövőben.

előre definiált [adat]típus (predefined [data] type) – →erősen típusos nyelvekben használható →[adat]típus, amelynek értékkészlete, ill. megengedett műveletei már eleve ismertek a konkrét →fordítóprogram számára (így nincs szükség explicit programozói definícióra). Rendszerint implementációfüggő része az illető programozási nyelvnek. – általában az →egész, a →Boole-, a →valós, ill. a →karakter [adat]típus. A klasszikus nyelvekben ismert típusok tkp. mind –ok, felhasználói [adat]típusok definiálására egyáltalán nincs lehetőség.

előregörgetés (roll-forward) – meghibásodott →adatbázis helyreállításának egyik módszere, ez az érintett rekordok aktualizálás utáni képét használja fel. Menete a következő: 1) a hiba felfedezése után betöltik az utol-

só →adatbázis-másolatot; 2) megkeresik a hibát közvetlenül megelőző →tranzakciót, vagyis azonosítják a meghibásodott részhalmazra vonatkozó aktualizálásokat; 3) az aktualizálás utáni képeket időrendi sorrendben visszatöltik az adatbázisba. Ha a hiba éppen aktualizálás közben következett be, úgy az utolsó tranzakciót újra fel kell dolgozni.

előtároló (cache memory) – kisméretű, gyors elérésű speciális memóriaegység, amely az információáramban a viszonylag lassúbb →operatív memória és a →processzor között helyezkedik el, s a memória elérési sebessége és a processzor műveleti sebessége közötti különbséget hivatott kiegyenlíteni. Az ~ban levő utasításokat többször is fel lehet használni a programban található →ciklusok és →szubrutinok miatt, s így nagymértékben csökken az operatív memóriából történő tényleges lehívások száma. (A magyar szakirodalomban gyorsítótár néven is szerepel.)

előtér-partíció (foreground partition) – az →operatív memória megosztásán alapuló →multiprogramozásnál az a →partíció, amelyben gyors választ, rövid végrehajtási időt igénylő, tehát magas →prioritással rendelkező programokat hajtanak végre. Az →osztott erőforrásokra vonatkozóan elsőbbséget élvez a →háttérpartícióban dolgozó programmal szemben. A legmagasabb prioritási szintet a →valós idejű üzemmódban dolgozó programok számára kell fenntartani.

előtérszín (foreground colour) – →bittérképezéses képernyőkezelésnél →szöveges módban a karakterek egyik alapvető attribútuma: magának a karakternek a színe (a másik két attribútum a →háttérszín és a →villogás).

előző függvény (precedent function) – a →felsorolt, ill. →sorszámozott [adat]típust használó programozási nyelvekben (pl. →Pascal, →Modula, →Ada stb.) az a →standard függvény, amely az →aktuális paraméterként megadott szimbólumhoz az eggyel kisebb sorszámú szimbólumot rendeli hozzá. Pl. ha a következő →deklarációt használták:

var Nyelv: (magyar, német, angol)

akkor az ~: **pred** (német) értéke „magyar" lesz.

elsőbbségi nyelvtan (precedence grammar) – →környezetfüggetlen nyelvtan, amelyben minden lehetséges szimbólumpárhoz legfeljebb egy elsőbbségi összefüggés tartozik, és nincs két olyan helyettesítési szabály, amelyeknek jobb oldala azonos. →Alulról felfelé haladó szintaktikai elemzést tesz lehetővé, vagyis az adott szöveg balról jobbra haladó olvasása során elsőbbségi összefüggések alapján lehet összevonni szimbólumokat egészen addig, amíg az eredeti szöveg a mondatszimbólumra nem redukálódik.

elsődleges betöltőprogram (bootstrap loader) – speciális rendeltetésű →betöltőprogram, amelynek feladata az →operációs rendszer rezidens részének feltöltése az →operatív memóriába. Elvileg két részből áll. Az első rész kizárólagos feladata a második résznek, a tulajdonképpeni ~nak a betöltése, és a →vezérlőpult speciális billentyűjének használatával kerül be az operatív memória fenntartott zónájába. Az első rész végrehajtásának eredményeként az operatív memóriába kerül az operációs rendszert tartalmazó →adathordozó (rendszerint →mágneslemez v. →mágnesszalag) első blokkja, amely végül is betölti az operációs rendszer rezidens részét.

elsődleges memória (main memory) – →operatív memória

első generációs számítógép (first generation computer) – kb. az ötvenes évek közepéig megvalósított, elektroncsöves áramkörökből felépített számítógép (kapcsolási sebessége milisec nagyságrendű), az elektronikus számítógépek első képviselője. Merev és viszonylag szegényes →architektúra jellemezte: késleltetővonalas v. elektrosztatikus memória, →periferikus művei között →nyomtató, →lyukkártyaolvasó, mágnesdob említhető. Igen nagy helyet foglalt el, és a disszipálódott hő elvezetése is komoly problémát jelentett, csak légkondicionált helyiségekben működött. Programozása szinte kizárólag →gépi nyelven volt lehetséges. A legelsőnek megépített, kizárólagosan elektronikus elemekkel működő számítógép, az ENIAC, 1946 és 1955 között üzemelt, kb. 18 ezer elektroncsövet tartalmazott. Technológiájukat, sebességüket és tárolókapacitásukat tekintve az ~ek ma már csak technikatörténeti jelentőségűek, bár a maguk idejében igen

komoly munkát végeztek, elsősorban katonai, atomenergia-kutatási, meteorológiai előrejelzési és más tudományos területeken.

első közelítés (first approximation) – valamely →iterációs eljárás kiindulási értéke. Sok numerikus módszer →konvergenciájának alapvető feltétele a megfelelően megválasztott ~.

eltolás (displacement) – a →relatív cím speciális megvalósítása. A →gépi utasítás →címrésze a programegység kezdetéhez viszonyított távolságát adja →bájtokban kifejezve. A →bázisregiszteres →áthelyezésben, az →átfedési szerkezetek megvalósításában játszik fontos szerepet, ugyanis az →effektív címet a jelzett bázisregiszter tartalmának és az ~nak az összege adja.

eltolt kitevő (biased exponent) – →karakterisztika

empirikus formula (empirical formula) – minden olyan összefüggés, amit valamilyen kísérleti megfigyelés, mérés útján nyert tapasztalati eredmények alapján állítanak fel. Sokszor kiinduló információja numerikus módszereknek, sőt egyesek éppen az ~kra támaszkodva keresnek mennyiségi összefüggéseket a vizsgált természeti jelenségekre.

emulálás (emulation) – létező v. még csak tervezett számítógéprendszer megvalósítása egy konkrét számítógép →erőforrásainak igénybevételével. Ez utóbbit bázisszámítógépnek, a megvalósítandót pedig célszámítógépnek is szokták nevezni. A bázisszámítógép rendszerint →mikroprogramozással v. →szoftverelemekkel szimulálja a célszámítógép →architektúráját.

entitás (entity) – számítástechnikai értelmezésben olyan elem (objektum), amelyről →információt tárolnak. Az ~nak különböző tulajdonságai lehetnek, s ezek mind rögzíthetők. Az adatfeldolgozás során gyakran előforduló fogalom még az ~halmaz, amelyet hasonló ~ok alkotnak, és amelyeknek a hasonló tulajdonságaikra vonatkozó információit rögzítették.

Az ~okkal kapcsolatban három értelmezési tartomány különböztethető meg. Az első az a valóság, ahol az ~ok léteznek s bizonyos tulajdonságokkal rendelkeznek. A második az az elvont értelmezés, amely a valóság vetülete az emberek (programozók) fejében. Itt az ~okra szimbolikusan hivatkoznak, s azok jellemzőiről beszélnek. A harmadik az adatok tartománya, ahol karakter- v. bitsorozatok segítségével kódolják az ~okra vonatkozó információkat.

erőforrás (resource) – valamely feladat elvégzéséhez szükséges eszköz. A számítástechnika szempontjából az ~okat két nagy csoportba sorolják. A hardver ~ok csoportjába tartoznak: a →központi vezérlőegység, az →operatív memória, a →háttértárak, a →periferikus művek stb., vagyis minden – a számítógéprendszer munkájában részt vevő – önálló hardvereszköz. A szoftver ~ok csoportjába tartoznak az →operációs rendszer keretében működő programok, amelyek részt vesznek a felhasználók programjainak kiszolgálásában és végrehajtásában.

Minden munka változó, de minden időpillanatban jól meghatározott ~igényt követel meg, és az operációs rendszer egyik alapvető feladata az ~okkal való gazdálkodás.

erőforrás-gazdálkodás (resource management) – az →operációs rendszer egyik legfontosabb funkciója, amelynek elsőrendű célja a rendelkezésre álló (hardver és szoftver) →erőforrások optimális kihasználása többfelhasználós környezetben. Magában foglalja az erőforrások jellemzőinek és foglaltsági állapotainak folytonos nyilvántartását, az erőforrásokra vonatkozó igények nyilvántartását és olyan kiszolgálását, hogy minél nagyobb legyen az erőforrások kihasználtsági együtthatója. A két legfontosabb hardver erőforrás az →operatív memória és a →központi vezérlőegység (ideje). Az ezekkel az erőforrásokkal folytatott gazdálkodás stratégiája adja az operációs rendszerek osztályozásának alapvető kritériumát (→monoprogramozás, →multiprogramozás, →időosztásos rendszer, →particionált memóriakezelés, →virtuális memória stb.).

erőforrás-kiosztás (resource allocation) – →multiprogramozásos rendszerekben használt stratégia a rendelkezésre álló →erőforrások (→központi vezérlőegység, →operatív memória, →periferikus művek) optimális kihasználására. Az ~ az →operációs rendszerek egyik alapvető feladata, lehet felhasználó-, folyamat- v. feladatközpontú. A kiosztási algoritmus jellege változó annak függvényében, hogy melyek az operációs rendszer optimalizálási kritériumai: az erőforrások opti-

mális kihasználása v. az egész rendszer átlagos válaszidejének minimalizálása.

Az ~ történhet centralizált módon, vagyis az összes rendelkezésre álló erőforrás szintjén v. decentralizált módon, az egyes erőforrások szintjén. Az első módszer globális felügyeletet biztosít a rendszer általános kihasználtsági fokának optimalizálására, míg a második az egyes erőforrások jellegének figyelembevételével azok speciális kihasználtsági fokának optimalizálását teszi lehetővé.

Más megközelítésben az ~ lehet statikus, amikor az erőforrások már eleve hozzá vannak rendelve a potenciális felhasználókhoz, v. dinamikus, amikor az ~ mindig a pillanatnyi viszonyoknak megfelelően váltakozva történik.

erőforrásosztás (resource sharing) – az →erőforrás-kiosztás olyan speciális megvalósítása, amely az adott →erőforrást látszólag v. ténylegesen egy időben több felhasználó rendelkezésére bocsátja. A látszólagos ~ jellegzetes megoldása az →időosztásos rendszer. Ennél a →központi vezérlőegység ideje az az erőforrás, amely meghatározott időszeletekre áll a különböző felhasználók rendelkezésére. Az →operatív memória ~sal működik a →lapkezelésnél, ahol ugyanaz a tényleges memória látszólag különböző programok lapjaihoz van egy időben hozzárendelve (természetesen csak látszólagosan, mert egy-egy új lap betöltése egyben a régi felülírását jelenti). Tényleges az ~ pl. az olyan →mágneslemeznél, amelyet egy időben használhat több felhasználó. Jellegzetes példája ennek az operációs rendszer nemrezidens részét hordozó mágneslemez, de felhasználói lemezegység is ki lehet osztva egyszerre több →partíció számára.

Az ~ vonatkozhat a szoftver erőforrásokra is. Ilyenek pl. az operációs rendszer adatkezelő szubrutinjai, amelyek →tiszta kódolásuk révén egyszerre több felhasználói programot is kiszolgálhatnak.

erősen típusos nyelv (strongly-typed language) – olyan magas szintű →programozási nyelv, amely megköveteli, hogy minden adatobjektumnak explicit módon deklarált →[adat]típusa legyen, s minden ilyen objektum csakis egy típushoz tartozzék. E megkötések egyben egyértelműen meghatározzák az adatok értékkészletét és a velük végezhető műveleteket. Az ~ →fordítóprogramja ellenőrzi, hogy az adatok használata összhangban van-e típusukkal (→típusegyezés), s így erőteljesen növelheti a programok megbízhatóságát. Az erős típusosság elvét a →Pascal vezette be, majd továbbfejlesztette a →Modula, a →CHILL és az →Ada.

érték (value) – programozástechnikai értelemben olyan attribútum, amely egy bizonyos objektumot képviselő adat állapotára vonatkozik. Nem szükségszerűen numerikus jellegű. Egyes programozási nyelvek lehetővé teszik nem numerikus értékhalmaz definiálását is, amelynek elemei jellemzői az adott objektum állapotváltozásainak. Pl. a →Pascal nyelvben deklarálható:

type Szín = (fehér, fekete, piros)

ahol a Szín objektum ~ei a jobb oldalon található halmazhoz tartoznak.

értékadó utasítás (assignment statement) – a legtöbb magas szintű →programozási nyelvben megtalálható →végrehajtható utasítás. Végrehajtásával valamely →változó →aktuális értéket kap. A változóhoz rendelt memóriában rendszerint aritmetikai v. logikai kifejezés kiértékelési eredményét tárolják. Az ~ magában foglalja a kifejezés kiértékelését, az eredmény átalakítását (ha megengedett) a megfelelő →[adat]típusra s az így nyert eredmény tárolását. Az új érték tárolása egyben a régi tartalom törlését is jelenti.

értékes számjegy (significant digit) – valamely szám kódolt formájában az a (rendszerint decimális) számjegy, amely még bizonyosan azonos az eredeti szám megfelelő számjegyével. A számítógépekben ugyanis a számok tárolására szolgáló memóriarekesz hossza adott és korlátozott, ami megszabja a számok pontosságát is. A kódoláskor nyert és tárolt jelsorozat tehát eltérhet annak – a választott kódban érvényes – korlátlan hosszúságú jelsorozatától.

érték szerinti hivatkozás (call by value) – →eljárások hívásánál (aktivizálásánál) használatos paraméterátadási módszer. Lényege, hogy az eljárás hívása előtt az →aktuális paraméterek értékeit átmásolják a →formális paraméterek számára fenntartott lokációkba. A hívott eljárásban tehát csak az aktuális paraméterek értékével lehet dolgozni, az azokhoz rendelt memóriarekeszek elérhetetlenek

a hívott eljárás számára, ezért az ~ nem alkalmas eredmények visszaadására a hívó eljárásba. A feldolgozandó objektumok számára a hívott eljárásban utalnak ki tárolóhelyet.

értelmezés (interpretation) – olyan módszer, amelynek segítségével valamely konkrét (v. absztrakt) számítógépen az illető →gépi nyelvtől eltérő kódolású programok végrehajthatók. Az ~ alapjában véve két fázisból áll: a végrehajtandó program elemzése, majd utasításonkénti végrehajtása (az adott gép utasításaival szimulálva a →forrásprogramban előírt tevékenységeket).

értelmezőprogram (interpreter) – magas szintű →programozási nyelven kódolt programok →értelmezésére szolgáló rendszerprogram. Szemben a →fordítóprogrammal, amely lefordítja a teljes →forrásprogramot és →tárgyprogramot állít elő, az ~ utasításról utasításra értelmezi a lefordítandó szöveget, s minden utasítást azonnal végrehajt. A szemantikai elemző nyújtotta információk alapján a gépre jellemző utasítások segítségével valósítja meg a magas szintű utasítás által meghatározott tevékenységet. Felépítését tekintve hasonlít a fordítóprogramhoz, amennyiben tartalmazza a →lexikális és a →szintaktikai elemzőt, de eltér tőle abban, hogy a →kódgenerálási fázis helyett végrehajtja az utasításban leírt funkciót.

érvényességi tartomány (validity region) – →blokkszerkezetű programozási nyelvekben (pl. →ALGOL, →PL/1, →Pascal, →Ada stb.), ill. →asszembler nyelvekben az →azonosítókhoz történő memóriarendelés, ill. az azonosítókra való hivatkozás kapcsolatára vonatkozó fogalom. Az ~ szempontjából a szimbólumok lehetnek →globális szimbólumok, ill. →lokális szimbólumok. A globális szimbólumok számára a definíciót tartalmazó programegységben rendelnek memóriát, s ezek bármelyik programegységből elérhetők →külső hivatkozás útján. A lokális szimbólumok csak a definíciót tartalmazó programegységből érhetők el, s természetesen ugyancsak itt kapnak memóriakiutalást.

ÉS (AND) – kétargumentumú →logikai függvény, amely akkor és csakis akkor igaz, ha mind a két argumentum egyszerre igaz. Igazságtáblázata:

ÉS	0	1
0	0	0
1	0	1

eszközvezérlő (device controller) – →vezérlő

F

F – a →hexadecimális ábrázolás jelkészletének eleme, amely a decimális 15 értéknek felel meg (bináris megfelelője: 1111).

fa (tree) – olyan →irányított gráf, amely eleget tesz a következő követelményeknek: 1) pontosan egy olyan csomópont van benne, amelybe nem vezet él (ezt a csomópontot gyökérnek nevezik); 2) minden csomópont elérhető a gyökérből irányított élek sorozatával; 3) a gyökér kivételével mindegyik csomópontba pontosan egy él vezet.

Kiegyensúlyozott egy ~, ha minden egyes csomóponthoz ugyanannyi számú számú ág tartozik, és a ~ elágazási kapacitása – kezdve a gyökértől, fentről lefelé haladva, minden sorban balról jobbra lépve – ki van használva. A legtöbb →keresési algoritmus a kiegyensúlyozott fastruktúrához illeszkedik. Ezek speciális fajtája, amelynek számítástechnikai szempontból külön jelentősége van, az ún. →bináris ~, ahol csomópontonként maximálisan két ág lehetséges.

fakszimile (facsimile) – számítástechnikai értelmezésben dokumentumok (szövegek, grafikus ábrák, képek stb.) olyan másolata, amelynél az eredeti letapogatása és a másolat előállítása közé →adatátviteli hálózat ékelődik. Ez azt jelenti, hogy a letapogatás útján előállított analóg v. digitális jeleket →kommunikációs csatornákon keresztül juttatják a rendeltetési helyre, ahol újra előállítják az eredeti dokumentumokat. Ma már nyilvános postai szolgáltatás →telefax néven.

fázis (phase) – számítástechnikai vonatkozásban egy időben és hatásában jól behatárolható feladatnak a számítógépes elvégzése. Pl. a →forrásprogramok →fordításában a következő alapvető ~ok különböztethetők meg: →lexikális analízis, →szintaktikai analízis, →szemantikai elemzés, →tárkiosztás, →kódgenerálás. Egy program számítógépre vitele is több ~on halad át: fordítás, →kapcsolás, →betöltés, tulajdonképpeni végrehajtás.

fejléc (header) – 1. →Adatátviteli hálózatokban használatos, az →üzeneteket, ill. az →adatcsomagokat kísérő járulékos információ. Vezérlési feladatokat lát el, segítségével az üzenetek irányíthatók. A feladó és a címzett azonosítóját, a hozzárendelt logikai csatorna nevét, több csomagra bontott üzeneteknél a csomagok sorszámát, esetleg a felhasznált →kódra és az üzenet tartalmára vonatkozó információkat tartalmazza. – 2. Programokban az allokációs információkat és a kezdő végrehajtási címet tartalmazó rész.

fejléckezdet karakter (start-of-heading character) – →szinkron adatátviteli eljárásokban használatos →vezérlőjel. Ha a lekérdezett →terminál →üzenetkapcsolásos üzemmódú, akkor ~rel kezdi üzenetét, s csak ezután adja vonalra a tényleges szöveget bevezető →szövegkezdés karakterét.

feladat (task) – 1. Olyan műveletek csoportja, amelyek önálló logikai egységet képviselnek az →operációs rendszer szempontjából. Egy-egy komplexebb problémát ~okra bontva oldanak meg, ez jelentős teljesítménynövekedést biztosít azáltal, hogy a rendelkezésre álló gépi →erőforrásokat dinamikusan lehet elosztani a különböző ~ok között. – 2. Egy program logikailag önálló egysége, amely ugyanannak a programnak más részeivel konkurens módon hajtható végre, vagyis elméletileg a →multiprogramozás elvének alapeleme ugyanannak a programnak a keretein belül. A programozó maga bontja ~okra programját, s ugyancsak ő írja elő azok aktivizálási időpontját, a szinkronizálási feltételeket, valamint a különböző ~ok közötti kommunikációt. Ezek a lehetőségek feltétlenül szükségesek bizonyosfajta felhasználásoknál (pl. a →valós idejű programozásban). Legfontosabb előnye, hogy valamilyen erőforrásra várva nem az egész program kerül megszakított állapotba, csak az erőforrást közvetlenül igénylő ~, s ezáltal lényegesen csökkenthető az egységes program végrehajtási ideje.

feladatkezelés (task management) – elsősorban a →párbeszédes üzemmódban dolgozó →operációs rendszereknek az egyik fontos funkciója. Hatáskörébe tartozik a →feladatok előállítása, nyilvántartása, aktivizálása, felfüggesztése, újraindítása, végrehajtási ütemezése, →erőforrásokkal történő ellátása, az egymással versenyhelyzetben lévő feladatok közötti konfliktusok megoldása, a feladatok közötti információcsere biztosítása stb. A ~ szempontjából szükséges infor-

mációkat a →feladatvezérlő blokkban gyűjtik össze, amely az illető feladatok előállításától kezdve azok megszűntéig a feladatok pillanatnyi állapotát tükrözi.

feladatvezérlő blokk (task control block) – a feladatorientált →operációs rendszerekben a feladatok vezérléséhez szükséges információk táblázatos gyűjteménye. Minden egyes feladathoz ~ van rendelve, s abban ilyen információkat tárolnak: a feladat azonosítója, →prioritása, a hozzárendelt →operatív memória határai, a feladat felfüggesztésekor a regiszterek aktuális tartalma, az utasításszámláló tartalma stb. A ~ alapvető eszköz a →multiprogramozás megvalósítására többfeladatos környezetben.

felbontás (resolution) – általános értelmezésben valamely grafikus képet felépítő →képpontok száma, a képminőség jellemzője.

feldolgozóegység (processing unit) – programozható digitális egység, amely önállóan v. az illető rendszerben található más elemekkel együttműködve képes a helyi feldolgozási igények kiszolgálására.

félduplex csatorna (half duplex channel) – az →adatátviteli hálózatokban használatos olyan →kommunikációs csatorna, amely csak felváltva, nem egyidejűleg, de mindkét irányba átviheti a hasznos adatokat és a vezérlőjeleket. Különösen középsebességű átviteli rendszerekben terjedt el, kiválóan alkalmas ember-gép kapcsolat létrehozására. A ~ egyszerű kéthuzalos összeköttetéssel valósítható meg.

félduplex összeköttetés (half duplex communication) – →adathálózat két →adatállomása közötti olyan kapcsolat, amelyben mindkét adatállomás felváltva küldhet és fogadhat adatokat, de egyidejűleg csak egy irányban áramolhatnak hasznos adatok. Az összeköttetés blokkolt állapotba kerülhet, ha mindkét adatállomás a másiktól várja az adatátvitelt. Értelmetlen állapotba is kerülhet a ~, ha mindkét adatállomás egyszerre küld adatokat, ugyanis a kapcsolat létrejöhet rendszertelenül beérkezett igények alapján is.

felejtőmemória (volatile memory) – olyan →memóriatípus, amely kikapcsolásnál v. feszültségkiesésnél elveszti tartalmát.

felfüggesztési pont (breakpoint) – végrehajtás alatt lévő program olyan pontja, ahol valamilyen külső hatás (→megszakítás, a →hibafelderítő program v. az →operációs rendszer közbelépése) következtében a szekvenciális végrehajtás felfüggesztődik. A felfüggesztést előidéző információ feldolgozása után a végrehajtás a sorrendben következő utasítással folytatódik. Nevezik még töréspontnak is.

felfüggesztett program (suspended program) – →multiprogramozásos rendszerekben végrehajtás alatt lévő olyan program, amely pillanatnyilag elvesztette a →központi vezérlőegység felhasználási jogát. Minőségileg abban különbözik a →megszakított program állapotától, hogy tkp. minden igényelt →erőforrás rendelkezésre áll, de →prioritása folytán még nem juthatott hozzá a központi vezérlőegységhez. Egy program felfüggesztett állapotba juthat akkor is, ha →ellenőrzött végrehajtás, ill. →lépésenkénti végrehajtás alatt áll.

felfűzéses hálózat (multipoint network) – →adatátviteli hálózatok olyan szervezése, amikor több, földrajzilag elkülönített →adatállomást sorban, egyetlen vonallal kötnek össze úgy, hogy a vonal az egyik adatállomástól a másikig vezet. Több ~ is csatlakozhat egyetlen központi számítógéphez. Mivel ugyanazon a vonalon két adatállomás egyidejűleg nem folytathat párbeszédet a számítógéppel, ez utóbbi sorban lekérdez minden adatállomást (→lekérdezéses szervezés). A ~ hátránya, hogy kevéssé megbízható, hiszen egyetlen →modem meghibásodása az egész vonalat használhatatlanná teszi, ugyanakkor a hiba behatárolása is elég körülményes.

felhasználó (user) – 1. Általános számítástechnikai megfogalmazásban olyan egyénre v. csoportra vonatkozó fogalom, aki v. amely jogosult a számítógéprendszer nyújtotta szolgáltatásokat igénybe venni. – 2. Az →operációs rendszer szempontjából minden olyan →feladat, program, amely igénybe veheti a rendszer által rendelkezésre bocsátott →erőforrások valamelyikét.

felhasználói azonosító kód (user identification code) – többfelhasználós, →párbeszédes üzemmódban dolgozó rendszerekben az egyes felhasználókat megkülönböztető kód. A ~ a →bejelentkezés elengedhetetlen információja.

felhasználói kézikönyv (user guide) – programtermékek kezelésére vonatkozó információkat tartalmazó dokumentáció, amelyet rendszerint az illető programtermékek előállítói bocsátanak a felhasználók rendelkezésére. Modernebb formája a →segélyállomány, főleg a →párbeszédes üzemmódban dolgozó rendszereknél.

felhasználói program (user program) – konkrét feladat számítógépes megoldására a számítógép üzemeltetője, bérlője v. annak megbízottja által előállított programtermék (pl. raktárnyilvántartás, üzemszervezés, helyfoglalási rendszer stb.). Nem azonos a számítógép gyártója által szállított programtermékekkel, bár azoknak bizonyos elemeit tartalmazhatja.

felsorolt [adat]típus (enumerated [data] type) – →erősen típusos nyelvekben használható, a programozó által definiálható elemi, →sorszámozott →[adat]típus. Értékkészletét a felhasználó szabja meg, felsorolva az összes lehetséges értéket. Ezek csak szimbólumok lehetnek, s felsorolásuk egyben sorrendjüket is meghatározza (→sorszámozott [adat]típus). A ~sal csak összehasonlítások, értékadások, ill. olyan →standard függvények végezhetők, mint a →sorszámfüggvény, az →előző függvény és a →következő függvény.

feltétel (condition) – valamely →program adatainak halmazán értelmezett predikátum, amelynek segítségével vezérelhető a különböző programrészletek végrehajtási sorrendje. A legtöbb magas szintű →programozási nyelvnek van olyan típusú →vezérlésátadó utasítása, amelynél a soron következő utasítássorozat bionyos ~ kiértékelési eredményétől függ.

feltételes elágazás (conditional branch) – döntési pont egy programban, bizonyos feltételtől függő →elágazás.

feltételes fordítás (conditional compiling) – egyes →fordítóprogramok speciális szolgáltatása. A fordítóprogram felismeri a jelzett programrészeket, s azokat csak feltételesen illeszti a forrásprogramba.

feltételes utasítás (conditional statement) – magas szintű →vezérlésátadó utasítás, amelynek vezérlő szerepe bizonyos feltétel kiértékelése nyomán hozott döntés révén valósul meg.

A klasszikus ~ két alapvető formája az

„IF feltétel THEN programrész-1 ELSE
programrész-2",
„IF feltétel THEN programrész".

A ~ általánosított változata többágú elágaztatást tesz lehetővé, formája:

„CASE kifejezés OF konstans-1: BEGIN
utasításlista END;
konstans-2: BEGIN
utasításlista END:
...
konstans-n: BEGIN
utasításlista END."

feltételes változó (conditional variable) – →COBOL nyelvben használt →változó, amelynek egy v. több értékéhez →feltételnevek tartoznak.

feltételes vezérlésátadó utasítás (conditional control instruction) – →gépi nyelvek, ill. →asszembler nyelvek rendszerint elmaradhatatlan utasítástípusa, amely bizonyos feltételtől függően megváltoztatja a program utasításainak végrehajtási sorrendjét. Alapvető jellemzője, hogy akkor és csakis akkor adja át a vezérlést az előre meghatározott →címre, ha az adott feltétel teljesül. Ellenkező esetben a program végrehajtása szekvenciálisan folytatódik.

feltételezett tizedespont (assumed decimal point) – numerikus adatmezőn belül két szomszédos karakter közé eső pozíció, amely nem tartalmazza ténylegesen a tizedespontot képviselő karaktert, de hasonló logikai jelentése van. A magas szintű programozási nyelvekben speciális leíró sablon létezik a ~ meghatározására. Pl. a →COBOL nyelvben a PICTURE IS 999V99 leírás olyan decimális számot ír le, amely három jegyet tartalmaz a tizedespont előtt az egész értékek számára és kettőt a tizedes résznek. Aritmetikai műveleteknél figyelembe veszik a ~ot.

feltételjegyzék (condition entries) – a →döntési táblázat jobb felső része: a feltételek lehetséges értékeiből közvetlen v. közvetett logikai műveletekkel képzett állapotok lehetséges változatai. Szavakkal kifejezett értelmezése: melyek az adott feltétel lehetséges értékei?

feltételnév (condition name) – a →COBOL nyelvben a →feltételes változó értékkészleté-

ből kiválasztott konkrét értékekhez v. értékintervallumokhoz rendelt →azonosító. Programozás szempontjából annak az eseménynek az elnevezése, amikor a feltételes változó aktuális értéke egyenlő a felsorolt értékek egyikével, v. benne van a felsorolt intervallumok valamelyikében.

feltételnév-feltétel (condition name condition) – →COBOL programozási nyelvben használt →egyszerű feltétel, amely szintaktikailag magából a →feltételnévből áll. A feltétel kiértékelésekor megvizsgálják, hogy a feltételnévhez tartozó →feltételes változó aktuális értéke egyenlő-e a feltételnévhez rendelt értékek egyikével, ill. benne van-e az adott intervallumok valamelyikében.

feltételtörzs (condition stub) – a →döntési táblázat bal felső része: a leírt rendszert jellemző elemek összessége, amelyek aktuális értékeikkel meghatározzák az illető rendszer (algoritmus) pillanatnyi viselkedését, ill. következő állapotát. Szavakkal kifejezett értelmezése: melyek a feltételek az adott pillanatban?

feltétlen vezérlésátadó utasítás (unconditional control instruction) – olyan utasítás, amely bármilyen feltételtől függetlenül megváltoztatja a program utasításainak végrehajtási sorrendjét. Mind a →gépi nyelvek (ill. →asszembler nyelvek), mind a magas szintű →programozási nyelvek utasításkészletében megtalálható.

felügyelő állapot (supervisor state) – →privilegizált állapot

felügyelőprogram (supervisor) – az →operációs rendszer központi vezérlőprogramja, amely az egész operációs rendszert irányítja.

Legfontosabb feladatai a következők: 1) figyeli és elemzi a →programmegszakításokat, és attól függően, hogy mi volt a →megszakítás oka (külső jel, bevitel v. kivitel kezdete v. vége, →hardverhiba, ismeretlen utasításkód stb.), a vezérlést annak a speciális szubrutinnak adja át, amely a megszakítás természetének megfelelően intézkedik; 2) vezérli a →multiprogramozást, azaz a →prioritások figyelembevételével gondoskodik arról, hogy az →erőforrások több feladat konkurens végrehajtásával lehetőleg teljesen ki legyenek használva; 3) irányítja az →operatív memória és a →periféria közötti adatforgalmat.

felügyelőprogram-hívó utasítás (supervisor call instruction) – olyan speciális utasítás, amely biztosítja a szoftver →erőforrások felhasználását. Amikor a →központi vezérlőegység ~t hajt végre, →csapda típusú megszakítást hoz létre (ismeretlen utasításkód alapján). A vezérlés automatikusan a megszakítást feldolgozó szubrutinra kerül, ezáltal biztosítva az igényelt művelet végrehajtását. Klasszikus példája a ~oknak a →lebegőpontos műveletek elvégzése, ha ezek nem tartoznak az →alapgép utasításkészletébe. A ~ végrehajtása a felhasználó szempontjából teljesen transzparens módon zajlik le, s így az alapgép utasításaival együtt mintegy a →kiterjesztett gép utasításkészletéhez tartozik.

felülről lefelé haladó módszer (top-down method) – →moduláris programozásban használható programfejlesztési stratégia. Elsősorban a tervezésben előnyös, de a kivitelezésben is alkalmazható. Ez azt jelenti, hogy a tervezés és – a kapcsolatok rögzítése mellett – a kivitelezés is az absztrakciós szintek sorrendjét követi. A ~ nagy előnye, hogy a program egésze már a fejlesztés korai szakaszában tesztelhető, viszont ehhez az alsóbb logikai szintű →modulokat szimulálni kell. Fontos előnye még az integrációs fázis teljes hiánya, a kapcsolási ütközések korai felfedezhetősége, az alacsonyabb logikai szintű modulok meghatározhatósága a már elkészült magasabb szintűek ismeretében, és így kevesebb a visszacsatolás és az újrakódolás.

A ~ hátránya az alacsonyabb szintű modulok szimulációjának problémája, valamint az, hogy lehetetlen a párhuzamos modultesztelés. Ezeket általában úgy lehet kiküszöbölni, hogy a ~rel végrehajtott tervezést az →alulról felfelé haladó módszerrel megvalósított kódolás és tesztelés egészíti ki egységes rendszerré (→strukturált programozás).

felülről lefelé haladó szintaktikai elemzés (top-down syntactical analysis) – →fordítóprogramok által a →szintaktikai analízis megvalósítására használt stratégia. Lényege, hogy a →szintaxisfát a gyökérelemből (kezdőszimbólumból) kiindulva a terminális elemek (vagyis az adott mondat szimbólumai) felé haladva építi fel. A részfák a leképzési lépések eredményei.

fényceruza (light pen) – a katódsugárcsöves grafikus megjelenítő tartozéka, ceruza alakú fényérzékelő eszköz, amely az ernyőn megjelenő kép adott pontjához hozzákapcsolja annak koordinátáit. Legfontosabb felhasználási területe a grafikus tervezés.

ferritgyűrű (ferrite core) – ferromágneses (általában vas-oxid alapú) anyagból készült, nagy koercitív erejű, gyűrű alakú vasmag. Kétállapotú, tehát egy →bit információ tárolására alkalmas elem. Ma már nem használják.

ferritmemória (core memory) – →ferritgyűrűkből mint elemi tárolókból felépített memória.

FIFO-szerkezet (FIFO-structure) – a számítástechnikában igen elterjedt szervezési technika. A szervezésben részt vevő elemek olyan láncolt rögzítését jelenti, amelynél új elemet elhelyezni a szerkezetben csakis az utolsó elem után lehetséges, ill. kiolvasni v. törölni mindig csak a legrégebben elhelyezett elemet lehet. A ~ alapvető tulajdonsága, hogy megőrzi a benne elhelyezett elemek időbeli sorrendjét. Jellegzetes alkalmazása a →várakozási sor.

figuratív konstans (figurative constant) – olyan speciális értékű konstans, amelyhez rögzített adatnév tartozik. Ez hozzátartozik az adott →programozási nyelv szókincséhez (→lefoglalt szó). ~ok pl. a →COBOL nyelvben: ZERO, ZEROS, ZEROES (programbeli szerepétől függően a nulla numerikus értéket v. egy-több zérus karaktert képvisel); SPACE, SPACES (egy v. több szóköz karaktert jelent); HIGH-VALUE, HIGH-VALUES (az adott gép karakterkészletében a legmagasabb értékűt képviseli); LOW-VALUE, LOW-VALUES (a karakterkészlet legalacsonyabb értékű karakterét jelenti).

firmver (firmware) – az angol szakirodalomból átvett →mikroprogramozási kifejezés. Rendszerint →szoftverfunkciókat ellátó, →permanens memóriákban tárolt mikroprogramokra utal. Pl. a decimális utasítások mikroprogramozott szimulációja olyan számítógépnél, amelyik nem tartalmaz decimális →aritmetikai egységet.

fixpontos művelet (fixed-point operation) – →fixpontos számábrázolású adatokkal végzett aritmetikai művelet (összeadás, kivonás, szorzás, osztás, →léptetés, összehasonlítás). Alapvető jellemzője, hogy végrehajtásakor az →aritmetikai egység nem veszi figyelembe automatikusan a tizedespont helyét, következésképpen a programozó feladata az operandusok és az eredmény helyes értelmezése.

fixpontos számábrázolás (fixed-point representation) – bináris számok megállapodás szerinti olyan ábrázolása, amely lehetővé teszi a →fixpontos műveletek elvégzését. Általános gyakorlat szerint a számokat fix hosszúságú szavakban tárolják, ahol a szó első bitpozíciója az →előjelbit. A ~ úgy valósul meg, hogy a tizedespontot nem ábrázolják fizikailag, hanem megállapodás szerinti fix helyet jelölnek ki számára. Ez a megállapodás előírhatja, hogy a tizedespontot közvetlenül az előjelbit után értelmezzék (ebben az esetben az összes ábrázolt szám abszolút értékben kisebb egynél), de előírhatja a tizedespontot az utolsó bitpozíció utánra is, ami azt jelenti, hogy a gépben ábrázolt számok mind egész számok.

fizikai [adat]állomány (physical file) – valamely →[adat]állománynak (mint konkrét adathalmaznak) valamilyen →adathordozón rögzített megvalósítása, →fizikai bejegyzéseinek szervezett összessége. A ~ lehet szerkezet nélküli adathalmaz, s akkor a programozónak kell fizikai szinten elvégeznie minden kezelési és feldolgozási műveletet. Ha a ~ szabványos logikai szerkezettel rendelkezik, akkor kezelése és feldolgozása is logikai szinten végezhető, az →[adat]állomány-kezelő rendszer megfelelő szubrutinjai segítségével. A →logikai [adat]állományokra jellemző szervezési technikát (→direkt szervezésű [adat]állomány, →indexszekvenciális [adat]állomány, →szekvenciális [adat]állomány) a különböző számítógéprendszerek sajátos ~nyal valósítják meg, amely rendszerint jellemző az illető rendszerre, de transzparens a logikai szintű feldolgozás szempontjából.

fizikai adatbázis (physical data base) – az →adatbázis globális leírásának (az →adatsémának) megfelelő adathalmaz fizikai megvalósítása valamilyen →adathordozón, vagyis az adatbázis összes →entitásának konkrét előfordulása. Az adathordozó szervezését és kezelését az →adatbázis-kezelő rendszer (az →operációs rendszer közreműködésével)

maga végzi el, biztosítva az →adatfüggetlenség minél magasabb szintjét.

fizikai adatszerkezet (physical data structure) – az az →adatstruktúra, amelyet egy adathalmaz az →adathordozón felvesz, és amelyet így a logikai adatszerkezet és az adathordozó sajátosságai együttesen határoznak meg.

fizikai bejegyzés (physical record) – az →adathordozóra jellemző ama adatmennyiség, amelyet egyetlen →b/k művelet segítségével cserélhetnek ki az →operatív memória és a →periferikus mű között (pl. egyetlen lyukkártya, egy →blokk a mágneslemeznél, ill. a mágnesszalagnál stb.). A ~ által elfoglalt hely az adathordozón két →blokk-köz közé esik a mágnesszalagon, ill. két címmező közé a mágneslemezen. A hatékony helykihasználás érdekében a ~ rendszerint több →logikai rekordból és esetleges járulékos adatból épül fel (→blokkolás). A legtöbb rendszerben a ~ hosszát a rendszerprogramozó határozza meg, bizonyos esetekben azonban fix hosszúságú. A ~ beolvasásához, ill. kiírásához megfelelő méretű →puffert kell kijelölni az operatív memóriában. A logikai rekordoknak ~ekbe szervezése, ill. a ~nek logikai rekordokká tördelése rendszerint az →[adat]állomány-kezelő rendszer feladata, s teljesen transzparens a felhasználó számára. (Kivételt csak a →szerkezet nélküli [adat]állomány jelent.)

fizikai cím (physical address) – az →abszolút címmel szinonim kifejezés, elsősorban a címezhető →adathordozókkal kapcsolatban használják.

fizikai kapcsolat (physical relationship) – a →fizikai adatbázis konkrét adat-előfordulásai között fennálló viszony, amit másodlagos adatok (pl. →mutatók, →indexek stb.) segítségével valósítanak meg.

fizikai rétegprotokoll (physical layer protocol) – →hétréteges referenciamodell

FLOPS (FLoating-point Operation Per Second) – →lebegőpontos aritmetikai egység műveleti sebességének mértékegysége (lebegőpontos művelet másodpercenként).

fogadójel (prompt) – végrehajtás alatt álló, →párbeszédes üzemmódban dolgozó →program által rendszerint a képernyőn közölt üzenet, amely azt jelenti, hogy a program kész új adatok, ill. utasítások fogadására. Sokszor csak a programra jellemző egyetlen karakter (pl. :, , $ stb.), de lehet olyan rövid szöveg is, amely a várt adatokra utal v. egyszerűen a program névjegye.

folyamatábra (flowchart) – →algoritmusok, →programok stb. grafikus ábrázolása szabványosított jelekkel. Előnyösen használható a rendszerek szerkezetének világos, érthető ábrázolására, a felépítő elemek között fennálló logikai kapcsolatok intuitív áttekintésére.

folyamatvezérlés (process control) – a számítógépek felhasználásának egyik igen fontos területe. Lényege, hogy a technológiai folyamatok vezérléséhez és szabályozásához szükséges döntéseket és a vezérlőjelek előállítását számítógéppel végeztetik. A ~t ellátó számítógépek rendszerint speciális kiépítettségű gépek (→folyamatvezérlő számítógép).

A ~nek két jól elkülöníthető fázisa van. Az első a folyamat ellenőrzése (figyelése), amelynek célja a mért adatoknak, ill. az elemzésük révén nyert eredményeknek, a folyamat állapotára vonatkozó adatoknak a megfelelő formában történő kijelzése v. rögzítése. A másik fázis a folyamat tényleges vezérlése a pillanatnyi állapot, ill. az előre rögzített célok megvalósítására szolgáló adatok alapján. Amikor e két fázis elkülönül, mivel közéjük emberi beavatkozás épül, nyílt hatásláncú ~ről beszélnek, ha viszont az ellenőrzés és a szabályozás nem igényel emberi beavatkozást, zárt hatásláncú a ~.

folyamatvezérlő számítógép (process control computer) – fizikai (technológiai) folyamatok irányítására, vezérlésére alkalmas számítógép. Többnyire →valós idejű feldolgozással működő, igen nagy biztonságú digitális számítógép. A ~nek analóg jelek érzékelésére és kibocsátására is alkalmasnak kell lennie, ezért →analóg-digitális, ill. →digitális-analóg konverterek kapcsolódnak hozzá. →Központi vezérlőegysége rendszerint igen hatékony →megszakítási rendszerrel és valós idejű órával, a mérőműszerek szolgáltatta nagyszámú paraméter fogadására pedig nagy hatásfokú →multiplexcsatornával rendelkezik.

Folyamatszabályozásra olykor →analóg számítógépet is használhatnak, de azt általá-

ban digitális számítógéppel →hibrid számítógéppé kötik össze.

fordítás (compiling) – az a komplex tevékenység, amelynek során valamely →programozási nyelven (→forrásnyelven) kódolt programot olyan →tárgyprogrammá alakítanak, amely közvetlenül végrehajtható egy konkrét v. absztrakt számítógépen. A ~t jellemző legfontosabb alapelv a →jelentésmegmaradás elve, amelynek értelmében a program jelentése nem változhat meg a rajta végzett (esetleg többszöri) átalakítás során.

A ~nak két alapvető feladata van: egyrészt a forrásnyelvű program helyességének ellenőrzése, az esetleges →szintaktikai, ill. →szemantikai hibák felismerése és kijelzése, másrészt a megfelelő tárgyprogram előállítása (generálása).

A tulajdonképpeni ~i tevékenység négy jól elkülöníthető fázisra tagolódik: elemzés (lexikális, szintaktikai és szemantikai), memóriakiosztás, kódgenerálás és tárgyprogramszerkesztés. Az elemzés során megvizsgálják a forrásprogramban szereplő nyelvi szerkezeteket, és összegyűjtik mindazokat az információkat, amelyeket a további fázisok majd felhasználnak (pl. a →szimbólumtáblázat). A memóriakiosztás során a szemantikai információk alapján kijelölik a szükséges tárterületet a forrásprogramban szereplő adatstruktúrák számára. A kódgenerálás fázisában az előzőekben felismert nyelvi elemekhez tartozó →tárgynyelvű (legtöbbször →gépi nyelvű) elemcsoportokat (utasítássorozatokat) állítják elő. Az utolsó fázis a programszerkesztés, amely a végleges tárgynyelvű programot, valamint a →kapcsolás, ill. a →betöltés elvégzéséhez szükséges járulékos információkat állítja elő.

fordítási egység (compilation unit) – valamely →programozási nyelven kódolt programszöveg, amelyet a →fordítóprogram egyszeri végrehajtása közben dolgoz fel. Lehet teljes, önálló program, de lehet annak csak töredéke (→szubrutin, →függvény, →eljárás, →modul, →csomag stb.) is. A ~re vonatkozó szabályokat alapvetően befolyásolja a programozási nyelv, ill. az annak megfelelő konkrét fordítóprogram (→külön fordítás).

fordítási hiba (compile time error) – valamely →forrásprogram fordítása során felismert hiba, amit a →fordítóprogram hibaüzenet formájában közöl a programozóval. A ~k több kategóriába sorolhatók, az egyszerű figyelmeztetésektől egészen a végzetes hibákig. A gyakorlatban rendszerint 4 hibaszintet határoznak meg.

1) Elsőfokú ~: figyelmeztetés, vagyis olyan egyszerű hiba, amit a fordítóprogram egyértelműen ki tud javítani, s így egyáltalán nincs befolyással sem a fordítás, sem a későbbi →kapcsolás, →betöltés, ill. végrehajtás menetére;

2) Másodfokú ~: olyan komolyabb hiba, amit a fordítóprogram úgy tud kijavítani, hogy több lehetséges változat közül választ egyet, s ez a felhasználó szempontjából v. megfelelő, v. sem;

3) Harmadfokú ~: olyan komoly hiba, amit a fordítóprogram nem tud kijavítani; a további fordításra nincs kihatással, de az eredményezett tárgyprogramot egyáltalán nem v. csak hibásan lehet végrehajtani;

4) Negyedfokú ~: végzetes hiba, a további fordítást is lehetetlenné teszi.

→Kötegelt feldolgozásnál rendszerint már a másodfokú hiba megakadályozza a további munkalépések (kapcsolás, betöltés, végrehajtás) elvégzését. Mégis, főleg a tesztelés időszakában, a felhasználó explicit módon kérheti ezek végrehajtását még harmadfokú hibánál is.

fordítási idő (compiling time) – a →fordítóprogram egyszeri végrehajtásának ideje.

fordítóprogram (compiler) – az egyik legfontosabb rendszerprogram, amely a →fordítás műveletét végzi el: a bemenetén jelentkező →forrásprogramot a kimenetén szolgáltatott →tárgyprogrammá alakítja át. Szerkezetét tekintve fázisokra és menetekre osztható. Az elvégzett műveletek természete szerint négy fázisra épül: →szintaktikai, ill. →szemantikai analízis, →memóriakiosztás, →kódgenerálás és →tárgyprogramszerkesztés. A ~ menetén a forrásprogram teljes átolvasását értik (forrásnyelven v. valamilyen közbülső formában). Egy-egy fázis annyira komplex lehet, hogy több menetet is igényel. Az →értelmezőprogramtól eltérően, a ~ az egész forrásprogramot lefordítja, előállítva a tárgyprogramot, amely v. a →kapcsolóprogram bemenete v. közvetlenül betölthető és végrehajtható.

fordítóprogram-író nyelv (compiler writing language) – →fordítóprogramok megvalósítására kifejlesztett magas szintű →programozási nyelv. Egyesíti magában a magas szint előnyeit az →asszembler nyelvek kedvező vonásaival. Legfontosabb nyelvi eszközei: az eljárások rekurzív hívhatósága, a bitszintű hozzáférés, a karakterláncok kezelhetősége, a cím típusú adatok feldolgozhatósága stb. A legismertebb –ek: →B, →BCPL, →C, →CDL, →Pascal stb.

fordított lista (push-down list) – speciális szerkezetű →lista: új adatelemet rögzíteni benne csak a mindenkori utolsó adatelem után lehet, kiolvasni v. törölni belőle csak az adatszerkezet utolsó adatelemét lehetséges. Tkp. a →veremmel azonos adatszerkezet.

fordított videó (reverse video) – képernyős →megjelenítőn kijelzett fontosabb adatelemek kiemelésére használt módszer. Lényege az, hogy a →háttérszínt és az →előtérszínt az illető adatelem által elfoglalt képernyőmező teljes területén felcserélik.

forgalmi mátrix (traffic matrix) – az →adatátviteli hálózatokat jellemző mátrix, amelynek (i,j) elemei az i-edik →csomópontból eredő és a j-edik csomópontnak szánt forgalom mennyiségét tartalmazzák. A mérési egység a hálózat fajtájától függően a másodpercenkénti hívások v. az →adatcsomagok száma.

formális nyelv (formal language) – minden mesterséges nyelv, amelynek →szintaxisát és →szemantikáját szigorú matematikai formalizmussal lehet leírni. A –re jellemző, hogy mondatai kötött szabályok szerint épülnek fel, egyszerűek és egyértelműen értelmezhetők.

formális paraméter (formal parameter) – hívott →eljárások algoritmusának megfogalmazására használatos programozási objektum. Ez a megfogalmazás független attól, hogy konkrét esetekben milyen paraméterértékek mellett történik az eljárás algoritmusának végrehajtása. A konkrét értékek átadása a hívó és a hívott eljárás között három alapvető módszer segítségével történhetik: →cím szerinti hivatkozással, →érték szerinti hivatkozással és →név szerinti hivatkozással.

formális rendszer (formal system) – értelmezés nélküli absztrakt matematikai kalkulus, amelyet rendszerint tudományos elméletek modellezésére alkalmaznak. Összetevői: 1) egy ábécé v. szimbólumhalmaz, úgy összeállítva, hogy bármilyen szimbólumról könnyen és egyértelműen meghatározható legyen, hozzátartozik-e az ábécéhez v. sem; 2) egy szóhalmaz, amely az ábécéhez tartozó szimbólumok olyan sorozataiból áll, amelyekkel az illető elmélet kijelentéseit modellezni lehet (axiómák); 3) egy relációhalmaz, amelyet kölcsönhatási szabályoknak neveznek. A számítástechnikában legelterjedtebben használt – a →Boole-algebra, a →Backus-Naur Forma stb.

A –ek fontos szerepet kaptak a programozási nyelvek tervezésében és implementációjában, elsősorban azok →szintaxisának és →szemantikájának leírásában. Felhasználásuk a szintaxisvezérelt →fordítással, a →fordítóprogramok helyességének bizonyításával és a nyelvek szerkezeti bonyolultságának vizsgálatával kapcsolatos.

formázás (formatting) – →mágneslemezek (→hajlékonylemezek, ill. →merevlemezek) felkészítése a logikai szintű felhasználás számára. A lemezeknél az egyes →sávok helyzete és a használható oldalak száma a gyártó által rögzített és megváltoztathatatlan. A sávokon belül a szektorok helyét, méretét és számát viszont →szoftverúton is rögzíthetik, vagyis a – révén kapják jellemző adataikat.

formulakezelő nyelv (formula manipulation language) – magas szintű →programozási nyelv matematikai kifejezések szimbolikus kezelésére. Legfontosabb felhasználási területe a fizikai és csillagászati kutatásokban felmerülő feladatok megoldása. Alapvető jellemzője a nagy pontosságú egész aritmetika, amely lehetőséget nyújt a programozónak, hogy tetszőleges pontosságú eredményekhez jusson. A legismertebb –ek között a FORMAC, a MACSYMA, a REDUCE stb. említhető.

forrásnyelv (source language) – az a →programozási nyelv, amelyen →fordítás, ill. →értelmezés céljából az eredeti programot írták. A magas szintű v. →asszembler programozási nyelveken kódolt programok közvetlenül nem hajthatók végre a rendelkezésre álló számítógépeken, ezért ezeket v. át kell alakítani (le kell fordítani) a konkrét számítógép →gépi nyelvére v. a konkrét gépi →utasításkészlet segítségével kell értelmezni.

forrásnyelvű könyvtár (source library) – →forrásnyelvű programok, ill. programegységek tárolására szolgáló →könyvtár. Legtöbbször mágneses →adathordozón található, s rajta – megfelelő rendszerprogramok segítségével – módosítások, javítások hajthatók végre. A →fordítóprogramok túlnyomó többsége közvetlenül a ~ból nyerheti a lefordítandó →forrásprogramot.

forrásprogram (source program) – valamely →programozási nyelv formalizmusával kódolt program, amely csak előzetes →fordítás v. →értelmezés útján hajtható végre. A ~okat rendszerint →forrásnyelvű könyvtárakban tárolják, karbantartásukra általában speciális rendszerprogramok szolgálnak.

„forró krumpli" adatútképzés („hot potato" routing) – olyan →adatútképzési módszer, amelynél, mihelyt lehetséges, →adatcsomagokat küldenek ki a →csomópontból, még akkor is, ha ez a kimeneti összeköttetés tekintetében rossz választásnak bizonyul. Ez a módszer, amelynek célja az adatcsomagok mindenáron való mozgásban tartása, néha oszcilláló hatásokat válthat ki: az adatcsomag két csomópont között ide-oda mozoghat, v. a csomópont körül, ahova irányították, végtelen ciklusba kerülhet.

FORTH – a hetvenes évek elején Charles H. Moore által kifejlesztett magas szintű →programozási nyelv, amely szakított a klasszikus programozási nyelvek filozófiájával. Egyesíti magában mindazokat az előnyöket, amelyeket a klasszikus magas szintű programozási nyelvek nyújtottak, ugyanakkor megvalósítja a →kiterjesztett nyelv fogalmát is. A ~ kiindulási magja egy sor standard definíció és egy olyan mechanizmus, amely lehetővé teszi a felhasználónak, hogy saját vezérlési és eljárási struktúráit az előbbiek segítségével meghatározza. Ilyenformán a kiindulási magból a konkrét felhasználásnak legjobban megfelelő, redundáns elemektől mentes programozási környezet alakítható ki. Magas szintű →hordozhatóságot biztosít azáltal, hogy a kiindulási mag és a kiterjesztési mechanizmus igen könnyen megvalósítható →operációs rendszer nélküli gépen is, ugyanis a magasabb szintű struktúrák már ~-definíciókban vannak leírva. Egyre nagyobb jelentőségre tesz szert a →mikroprocesszoros fejlesztéseknek, a robotok megvalósításának köszönhetően.

FORTRAN – →eljárásra orientált magas szintű →programozási nyelv, egyike a legrégibb és legelterjedtebb programozási nyelveknek. Első változatát 1956-ban készítették el, s azóta több standardizált változata terjedt el: FORTRAN II, FORTRAN IV, FORTRAN 77 stb. Főleg tudományos-műszaki adatfeldolgozásra alkalmas, de sok más területen is használható. Feladatorientáltsága elsősorban abban nyilvánul meg, hogy kevésbé alkalmas nagy tömegű adathalmazok mozgatására és feldolgozására (bár bizonyos →[adat]állomány-kezelési lehetőségeket azért tartalmaz), viszont igen gazdag a matematikai függvényválasztéka. Modularitása alkalmassá teszi a →moduláris programozás, ill. →strukturált programozás elvei szerinti programfejlesztés bizonyos fokú megvalósítására. E képessége még növelhető egyes hiányzó vezérlési struktúrák (pl. WHILE, REPEAT, FOR stb.) beépítésével, amelyeket →előfordítás útján a már meglévő eszközökkel lehet megvalósítani. Pl. az →UNIX →operációs rendszer komponense, a RATFOR előfordító a következő módon építi be a FOR vezérlési szerkezetet:

```
[címke] FOR (utasítás-1; feltétel; utasítás-2)
                                    utasítás-3
```

amely fordítás után a következő formára alakul át:

```
[címke]   utasítás-1
  3       IF (.NOT.(feltétel)) GO TO 2
          utasítás-3
  1       utasítás-2
          GO TO 3
  2       CONTINUE
```

→[Adat]típusait tekintve a ~ nem túl gazdag, csupán az egész, a valós, a komplex és a logikai [adat]típusokat ismeri, egyedüli adatstruktúrája a homogén →tömb. Ennek ellenére az általánosan használt programozási nyelvek között előkelő helyet foglal el, szinte nincs általános célú operációs rendszer, amely ne rendelkezne ~ →fordítóprogrammal.

főindex (master index) – →indexszekvenciális [adat]állományoknál a legmagasabb indexszint neve (a közvetlenül alatta található

szint a →cilinderindex). A ~ rendszerint a nagy [adat]állományoknál található meg, s a cilinderindexben való keresést gyorsítja meg.

főprogram (main program) – a →moduláris programozásban használatos fogalom: programoknak az a része, amely a különböző funkciójú programegységek vezérlését, logikai sorrendi aktivizálását hivatott megvalósítani. A programok →betöltése mindig a ~mal indul, s rendszerint az illető program teljes végrehajtása alatt az →operatív memóriában található. A ~ aktivizálásával kezdődik az adott program végrehajtása, közvetlenül az operatív memóriába való betöltés után.

frissítés (refreshing) – általános technikai értelmezésben elektromos v. mágneses állapot periodikus visszaállítása, regenerálása (→képfrissítés, memóriafrissítés).

„front-end" számítógép (front-end computer) – →kommunikációs processzor

funkcionális billentyű (functional key) – a →billentyűzet speciális kategóriájába tartozó billentyű: lenyomása egy-egy meghatározott művelet elvégzését váltja ki. A legfontosabb ~k, amelyek a legtöbb →terminál tartozékai: a →helyőr mozgatását, a tabulálást, a soremelést végző, a karaktertörlő, a visszaléptető stb. billentyű. A legtöbb billentyűzet olyan ~ket is tartalmaz, amelyeket a felhasználó saját igényeinek megfelelően maga is programozhat.

funkcionális függőség (functional dependency) – a →relációs adatbázisokkal kapcsolatos fogalom: egy adattétel ~ben van egy másikkal, ha az utóbbi az előzőt egyértelműen meghatározza. Más megközelítésben ~ áll fenn akkor, ha egy tulajdonság értékét egy másik tulajdonság értéke egyértelműen meghatározza. Pl. egy város neve egyértelműen meghatározza (általában) a megye nevét is.

futószalagszerű feldolgozás (pipeline processing) – a számítógépes adatfeldolgozás speciális módszere. Lényege, hogy a bemenő adatokon elvégzendő komplex átalakítást nem egy funkcionális egység (program) végzi el teljesen, hanem több egyszerű részműveletre bontva, szekvenciálisan, több feldolgozó egységgel végeztetik el. Az egyes részműveletek átfedhetik egymást, így a feldolgozás hatékonyabbá tehető. Hátránya, hogy kiegészítő, szinkronizáló és ütemezési tevékenységeket is be kell iktatni.

futtatás (running) – valamely programtermék által meghatározott feladatok elvégzése adott számítógéprendszer segítségével. A ~ minimálisan két fázisból áll: a →betöltésből és a tulajdonképpeni végrehajtásból, de tartalmazhat olyan előkészítő fázisokat is, mint a →fordítás, →kapcsolás stb.

független feltételrendszer (independent condition system) – a →döntési táblázatban olyan eset, amikor egy-egy szabályon belül az adott feltételek közül csak egynek van értékes bejegyzése, de a táblázatban többféle állapotuk is fellép. Pl.

A ≤ B	Y	N	–	–
C > D	–	–	Y	N

függő feltételrendszer (dependent condition system) – a →döntési táblázatban olyan eset, amikor minden feltétel minden állapotával a többi feltétel összes állapota előfordul. Pl.

Házas	Y	Y	N	N
Nő	Y	N	Y	N

függvény (function) – programozástechnikai értelmezésben az →eljárások olyan típusa, amelynél az azonosító névnek kettős funkciója van: egyrészt vele lehet a ~t aktivizálni (végrehajtani), másrészt a ~ neve egyben →változó is, amelynek értékét maga a ~ állítja elő, végrehajtása során. Ilyenformán a ~ operandusként vehet részt az aritmetikai v. logikai kifejezésekben.

fürt (cluster) – 1. Egy közös vezérlő (pl. →multiplexcsatorna) irányítása alatt álló →periferikus művek csoportja. – 2. →Kommunikációs csatornához egy pontban csatlakozó →adatállomások csoportja.

füzér [adat]típus (string [data] type) – →erősen típusos nyelvekben alfanumerikus karakterláncok kezelésére definiált →strukturált [adat]típus. A gyakorlatban többfajta ábrázolási módszert alkalmazhatnak. A leggyakoribb az →ASCIIZ-füzér, ill. a füzér első →bájtját használják a tényleges karakterszám rögzítésére.

G

G – számítástechnikai vonatkozásban mértékegységek (pl.tárolási kapacitás) kifejezésére használt egyezményes jel, a giga rövidítése. Értéke: 1**G**egység = 1024 →**M**egység = 1024 x 1024 →**K**egység = 1024 x 1024 x 1024 egység = 10 773 741 824 egység.

gazdaszámítógép (host computer) – →adatátviteli hálózatok központi számítógépe, amely a helyi szolgáltatások ellátásán kívül megvalósítja a →hálózatvezérlési funkciókat is.

generációszám (generation number) – négyjegyű decimális szám, amely lehetővé teszi egy →[adat]állomány azonosítását két v. több azonos nevű [adat]állomány között. Szabványos helye a HDR1 →[adat]állománykezdőcímkében van. Jelentősége a →könyvtárkezelő program szempontjából van, ugyanis így egyazon programnak v. [adat]állománynak több változatát is tárolni lehet ugyanabban a →könyvtárban.

generatív karakter (wildcard character) – egyes →operációs rendszerekben (ill. →[adat]állomány-kezelő rendszerekben) használható olyan karakter, amely egész [adat]állománycsoportot azonosító →[adat]-állománynév meghatározására alkalmas. Lényegében kétféle ~ használható. A „?" ~ egyetlen, a „*" ~ tetszőleges számú karaktert helyettesíthet. (A magyar, ill. a német szakirodalomban „dzsóker" karakterként is említik, a kártyajátékokból vett analógia nyomán.)

generatív nyelvtan (generative grammar) – olyan →formális rendszer, amely képes véges definícióját adni egy végtelen →nyelvnek, vagyis olyan szabályok rendszere, amelyeknek alapján az adott nyelvhez tartozó bármely mondat előállítható. A ~ tkp. a matematikai logikában használatos kalkulusok speciális fajtája.

Formális definíciója szerint a ~ olyan (T,N,S,P) négyes, ahol a T a →terminális szimbólumok halmaza, az N a →nemterminális szimbólumok halmaza, az S a kezdőszimbólum, a P pedig a →helyettesítési szabályok véges halmaza. A helyettesítési szabályokra vonatkozó megszorítások alapján Noam Chomsky négy nyelvtípust határozott meg (→Chomsky-hierarchia).

generikus egység (generic unit) – az →Ada programozási nyelvben használható programozási egység (→alprogram v. →csomag), amely mintaként szolgál egy egész eljáráscsaládnak. Ez a minta közvetlenül nem hajtható végre, de a végrehajtási időben átvett →aktuális paraméterek révén olyan példányai jönnek létre, amelyek rendelkeznek a modell összes tulajdonságával, mégis a konkrét paramétereknek megfelelően működnek.

gépfüggetlen (machine independent) – olyan programokra, eljárásokra vonatkoztatott jelző, amelyeknek leírásában, megvalósításában nem használták fel bizonyos konkrét számítógéprendszer speciális lehetőségeit. Az ilyen eljárások tervezésében csupán a megoldandó feladat logikai összefüggéseit, a különböző fázisokban elérendő állapotokat, a be-, ill. kimenő adatok természetét, formáját, értékintervallumait stb. írják le, anélkül hogy a megvalósítás konkrét eszközeit előírnák. A leírás történhet →folyamatábrákkal, →döntési táblázatokkal, esetleg olyan magas szintű →programozási nyelvekkel, amelyeknek standard változatai rendszerint a legtöbb számítógépen működnek (pl. →BASIC, →FORTRAN, →COBOL, →PL/1, →Pascal stb.).

gépi ciklus (machine cycle) – az az időintervallum, amely a →központi vezérlőegység két regisztere közötti adatátvitelhez szükséges. Egy bizonyos számítógép feldolgozási sebességének egyik fontos jellemzője.

gépi fordítás (machine translation) – természetes nyelveken írott szövegek lefordítása ugyancsak természetes nyelvre, számítógép segítségével. Bár a számítógépek ilyen célú felhasználásának ötlete már az ötvenes évek elején felmerült, gyakorlatba ültetése csak 1960 után, a szimbólumfeldolgozó nyelvek megjelenésével egy időben kezdett bizonyos eredményeket felmutatni. A ~nál lényegében négy fő feladatot kell megoldani: a gépi szótárak létrehozását, a forrásnyelvű szöveg →szintaktikai és →szemantikai elemzését, a strukturális átalakítást, vagyis a két nyelv közötti szerkezeti és szórendi különbségek áthidalását, a tárgynyelvű szöveg szintézisét. Érdekes irányzatként említhető az eszperantónak mint közbenső (átmeneti) nyelvnek alkalmazása a legújabb fordítási rendszerekben.

gépi kód (machine code) – azoknak a szabályoknak az összessége, amelyek meghatározzák a →gépi utasítást felépítő információk olyan ábrázolását, hogy azt a számítógép →központi vezérlőegysége közvetlenül végre tudja hajtani. A gépi utasításszó az adott gépre jellemző mezőkre oszlik, s ezeket kell olyanformán kódolni, hogy azok a központi vezérlőegység számára értelmes információkat jelentsenek.

gépi nyelv (machine language) – géporientált →programozási nyelv, amely a konkrét gép →utasításkészletének és a →gépi kód szabályainak felhasználásával az illető gépen közvetlenül (tehát →fordítás, ill. →értelmezés nélkül) végrehajtható programok írására (előállítására) alkalmas. A számítógépek programozásának legalacsonyabb szintje, s alapjában véve azt jelenti, hogy a számítógép →operatív memóriáját valamilyen módon úgy töltik fel, hogy ezeket a bináris információkat a gép utasításokként tudja értelmezni. A ~ programozásra ma már csak különleges esetekben használatos, ismerete viszont elengedhetetlenül szükséges a szoftverfejlesztés területén dolgozó programozóknak (elsősorban a →fordítóprogramok, →értelmezőprogramok, →operációs rendszerek és más rendszerprogramok előállításánál, hiszen a forrásnyelvű szerkezetekhez ~ű sorozatokat kell rendelni).

gépi utasítás (machine instruction) – valamely számítógép →központi vezérlőegysége által közvetlenül értelmezhető és végrehajtható utasítás. Hossza, felépítése és értelmezése az adott központi vezérlőegység megannyi alapvető jellemzője (→gépi kód).

gépkezelői beavatkozás (operator's intervention) – →kötegelt feldolgozással működő →operációs rendszereknél használt →megszakítás a gépkezelő és a rendszer közötti dialógus kezdeményezésére. Rendszerint a →vezérlőpult speciális billentyűjének kezelésével jön létre a megszakítás, amely a dialógust vezérlő rendszerszubrutinnak adja át a vezérlést (minden más program végrehajtása megszakad). A dialógus olyan szervezési műveletek végrehajtását teszi lehetővé, amelyek a kötegelt feldolgozás körülményei között az →erőforrások optimális kihasználását biztosítják.

géppel feldolgozható forma (machine processible form) – adatok olyan kódolása és rögzítése, hogy azokat közvetlenül feldolgozhassa egy önműködő adatfeldolgozó rendszer. A ~ általában egy fizikai (optikai, mágneses stb.) mennyiség változásával jellemzi az adatokat (→lyukkártya, →lyukszalag, →mágnesszalag, →mágneslemez stb.).

globális hiba (global error) – olyan →hiba, amely az adott program →fordításának, →kapcsolásának v. végrehajtásának egész kimenetelére hatással van.

globális szimbólum (global symbol) – egy program valamely egységében (→modul, →blokk, →szubrutin stb.) definiált →azonosító, amely a többi egységből →külső hivatkozás útján érhető el. Más megközelítésben olyan azonosító, amely egy adott program teljes egészében egyértelműen ugyanarra az objektumra vonatkozik. A ~okra történő hivatkozások a program →kapcsolásakor v. →betöltésekor oldódnak fel. Egy szimbólum globális jellegét v. explicit →deklaráció, v. az erre vonatkozó implicit →szintaktikai szabály biztosítja. Pl. a →blokkszerkezetű programozási nyelveknél (→PL/1, →Pascal stb.) minden külső blokkban deklarált szimbólum a belső blokk számára implicit ~.

globális szimbólumtáblázat (global symbol table) – →fordítóprogramok (ill. →asszemblerek) által előállított táblázat, amely járulékos információkat tartalmaz a →kapcsolóprogram v. a →betöltőprogram számára a →külső hivatkozások feloldására.

görgetés (scrolling) – →képernyőn megjelenített szöveges információ mozgatása függőleges v. vízszintes irányban. Történhet a képernyő egész felületén v. csak egy-egy →ablak keretein belül, soronkénti (oszloponkénti) ugratással v. folyamatosan.

gráf (graph) – matematikai objektum, amely geometriailag csomópontokból és az ezeket összekötő élekből áll. Az élek a csomópontok között fennálló esetleges kapcsolatokat fejezik ki. Az irányított ~nál nemcsak az jellemző a két csomópont kapcsolatára, hogy össze vannak-e kötve v. sem, hanem az is, hogy az esetleges él melyik csomópontból indul ki. A számítástechnikában különleges jelentőségű irányított ~ a →fa.

grafikus adatfeldolgozás (graphic data processing) – →számítógépes grafika

grafikus alakfelismerés (graphic pattern recognition) – grafikus információk feldolgozására szükséges módszerek és eszközök összessége. A grafikus információ →digitalizálása nyomán keletkezett digitális információelemek önmagukban még nem képesek kifejezni a képi információ jelentéstartalmát. Komplex eszközök és programok szükségesek ahhoz, hogy a képi információelemek digitális átalakítása mellett együttes jelentésüket is kódolni lehessen. Tipikus felhasználási területe a számítógéppel segített térképészet, tomográfia, állapotelemzés, a kézírásos szövegek felismerése és a robottechnika.

grafikus formák generálása (graphic pattern generation) – a grafikus adatfeldolgozás fontos alkalmazási területe. Központjában egy olyan →adatbank található, amely grafikus és nemgrafikus elemek (pontok, egyenes szakaszok, körívek, körök, sík és térbeli görbék, felületek generálására alkalmas →szubrutinok) gyűjteménye. Ezekből az elemekből – meghatározott célok megvalósítására – v. a felhasználói program, v. a grafikus →terminál előtt ülő felhasználó parancsai alapján komplex alakzatok állíthatók elő, esetleg időben változó formában is. Ilyen módon bonyolult tervezési és szimulációs feladatok oldhatók meg, de felhasználható rajzfilmek, reklámfilmek készítésére, dekorációs célokra stb. is.

grafikus kijelzés (graphic displaying) – számítógéprendszer által szolgáltatott, grafikus formában (függvénygörbék, táblázatok, diagramok stb.) megjelenő információ.

grafikus kijelző (graphic display) – az →interaktív (számítógépes) grafika alapvető eszköze, amely a számítógépes program által generált adatokat a megfelelő gyorsasággal grafikus információkká alakítja át, és képernyőn megjeleníti. Két alapvető típusa a →vonalrajzoló és a →raszteres grafikus kijelző. A számítógéprendszerben elfoglalt helye alapján a ~ olyan speciális →periferikus mű, amely a →központi vezérlőegység felügyelete alatt dolgozik és →közvetlen tárhozzáférést használhat.

grafikus mód (graphic mode) – →képernyők v. →nyomtatók egyik alapvető üzemmódja, amelynél a képet felépítő minden képpont külön vezérlés alatt áll. A ~ segítségével elsősorban bonyolult rajzok állíthatók elő, de alkalmas különböző formájú és méretű →karakterek megjelenítésére is (másik üzemmód a →szöveges mód).

grafikus terminál (graphic terminal) – grafikus ábrák (függvénygörbék, diagramok stb.) megjelenítésére alkalmas, →katódsugárcsöves kijelzővel v. →rajzgéppel ellátott →terminál. Ha a képernyős típus ábraelemek bevitelére alkalmas →fényceruzával is rendelkezik, párbeszédes ~nak is nevezik.

grammatika (grammar) – →nyelvtan

GY

gyorsítótár (cache memory) – →előtároló

gyorsrendezés (quicksort) – az egyik leghatékonyabb →belső rendezési módszer. Lényege, hogy az eredeti adathalmazt két részre osztják: az egyikbe kerülnek azok az elemek, amelyek a halmazból választott tetszőleges elemnél kisebbek, a másikba pedig azok, amelyek nem kisebbek. Ugyanezt a felosztást alkalmazzák a két részhalmazra vonatkoztatva is, majd így tovább, amíg minden egyes részhalmaz csupán egy elemből áll.

gyökérszegmens (root segment) – az →átfedési szerkezettel rendelkező programnak az a része, amely az első végrehajtandó utasítást tartalmazza, s a program egész végrehajtása alatt az →operatív memóriában található. A magas szintű →programozási nyelveknél a →főprogramnak felel meg.

H

hajlékonylemez (floppy disk) – mágnesezhető anyaggal bevont kisméretű, hajlékony műanyag korong, amelynek egyik v. mindkét oldalán információt lehet tárolni. Első jellegzetes képviselője az IBM szabvány: átmérője 8 hüvelyk, 76 sávot, egy sáv 27 szektort, egy szektor pedig 128 →bájtot tartalmaz. Elterjedt változatai az 5.25, ill. 3.5 hüvelykes ~. Tárolási sűrűsége lehet: egyszerű (SD) 2768 →bpi, dupla (DD) – 5536 bpi és nagyfokú (HD) – 9640 bpi sűrűséggel. Tárolási kapacitása kb. 250 Kbájt és 2 Mbájt között, hozzáférési ideje 100-250 msec, olvasási sebessége 100-200 Kbájt/sec között váltakozik. Nagy jelentősége van a mikro- és miniszámítógépes alkalmazásokban mint legfontosabb →adathordozónak. Az új szabvány a 3.5 hüvelykes, 2.88 Mbájtos ~ (IBM PS sorozat).

halmaz [adat]típus (set [data] type) – →erősen típusos nyelvekben definiálható, rögzített számú, egyazon →sorszámozott [adat]típusú adat együttes kezelésére használható →strukturált [adat]típus. Tkp. a matematikából ismert halmaz fogalmának átültetése a programozási nyelvekbe, így megengedett műveletei is (az értékadás mellett) jellemzőek: egyesítés, keresztmetszet, hozzátartozás, egyenlőség- és részhalmazvizsgálat.

hálózati rétegprotokoll (network layer protocol) – →hétréteges referenciamodell

hálózatvezérlés (network control) – az →adatátviteli szoftver egyik alapvető feladata, amelyhez a következő legfontosabb tevékenységek tartoznak:

1) A hálózat vonalainak fizikai vezérlése, amelynek során az →üzeneteket, ill. →adatcsomagokat kísérő vezérlő karakterek is átvitelre kerülnek;

2) A →terminálok logikai és fizikai címei közötti kapcsolat megteremtése, a logikailag kijelölt terminálok kiválasztása;

3) Az üzenetekhez rendelt →pufferek kezelése, dinamikus láncolása;

4) Az adatátviteli eljárások kiválasztása és elvégzése a felhasználó →makroutasításai alapján (a rendszer által generált →csatornaprogramok segítségével).

hangolás (tuning) – általános értelmezésben azon módosító műveletek összessége, amelyeket adott komplex rendszer részegységein hajtanak végre az egységes rendszer általános teljesítményének optimalizálása céljából. Számítástechnikai vonatkozásban különös jelentősége van a már kész programtermékek utólagos módosításaiban, amikor a környezeti körülmények változásai azt megkövetelik (→rugalmasság, →adatbázis-hangolás, →adatvándorlás).

hardver (hardware) – az angol szakirodalomból átvett általános jellegű szakkifejezés valamely számítógéprendszer összes műszaki (mechanikus, mágneses, optikai, elektromos stb.) berendezéseire (megkülönböztetésül a számítógéprendszerhez tartozó összes programterméktől, amelyek közösen a szoftver elnevezést viselik).

harmadik generációs számítógép (third generation computer) – kb. 1965 és 1972 között →integrált áramkörös elektronikával épült →digitális számítógép (kapcsolási sebessége nanosec nagyságrendű). Alapvető jellemzője a gyors, viszonylag olcsó és megbízható →architektúra. Ennél a generációnál jelentek meg a közvetlen hozzáférésű, igen nagy kapacitású, rendszerint mágneslemezes →háttértárak (5-25 Mbájt). A fejlett, egyre bonyolultabb →hardver gazdaságos kihasználása, ill. vezérlése már nem valósítható meg az →operációs rendszer közreműködése nélkül. Szervezési szempontból legfontosabb újítása a →multiprogramozás és a →párbeszédes üzemmód (→számítógép-generáció).

határregiszter (limit register) – →particionált memóriakezelésnél az egyes →partíciók védelmét biztosító eszköz. Egy program bármelyik hivatkozása, ami a ~ek által határolt zónán kívülre vonatkozik, azonnal →megszakítást eredményez, és az adott program végrehajtása rendellenesen befejeződik.

háttérfeldolgozás (background processing) – →párbeszédes üzemmódban dolgozó rendszerekben a felhasználó beavatkozását nem igénylő munkák elvégzése. Ezeket a feladatokat a felhasználó csak bejelenti a rendszernek, tényleges végrehajtásukra automatikusan, az igényelt →erőforrások rendelkezésre állásakor kerül sor, addig a rendszer által kezelt →várakozási sorban találhatók. Jellegzetes ~ pl. egyes →[adat]állományok (aszinkron) nyomtatása.

háttérnyomtatás (spooling) – →[adat]állományok aszinkron nyomtatását végző →háttérfeldolgozás.

háttérpartíció (background partition) – az →operatív memória megosztásán alapuló →multiprogramozásnál olyan →partíció, amelyben alacsony →prioritású, könnyen félbehagyható, láncolt munkalépésekből álló programok dolgoznak. Elsősorban az olyan rendszerprogramok végrehajtási helye, mint a →fordítóprogramok, a →kapcsolóprogramok, a →betöltőprogramok, a →könyvtárkezelő programok, a →rendezőprogramok stb., tehát amelyeknek válaszideje nem kritikus sem a felhasználó, sem a rendszer szempontjából. A ~ban tartózkodó program csak addig futhat, amíg az összes →előtér-partíció valamilyen okból kifolyólag felfüggesztett v. megszakított állapotban van (pl. →b/k művelet befejezését várja). A ~ rendszerint a →kötegelt feldolgozás dedikált partíciója.

háttérszín (background colour) – →bittérképezéses képernyőkezelésnél, →szöveges üzemmódban a karakterek egyik alapvető attribútuma: a karakter által el nem takart terület színe (a másik két attribútum az →előtérszín és a →villogás).

háttértár (secondary memory) – azoknak a számítógéphez csatolt, adatok tárolására alkalmas berendezéseknek a közös elnevezése, amelyekhez a →központi vezérlőegységnek nincs közvetlen hozzáférési lehetősége. A ~ban tárolt adatok speciális utasítások (→csatornaparancsok) segítségével vihetők be az →operatív memóriába, s csak azután állanak a feldolgozás rendelkezésére. Legismertebb típusai: mágnesdob, →mágneslemez, →mágnesszalag, →hajlékonylemez stb.). A ~ legfontosabb jellemzői: nagy tárolási kapacitás, viszonylag alacsony tárolási költség, viszont az operatív memóriáénál sokkal nagyobb hozzáférési idő. Egyes operációs rendszerek az operatív memóriából és a ~ból többszintes →memóriahierarchiát hoznak létre, ahol a különböző szintek közötti →adatvándorlás a felhasználó szempontjából transzparens módon zajlik le.

házi számítógép (home coputer) – háztartásban, a szabadidő kitöltésében, személyes otthoni használatra szánt →mikroszámítógép. Jellegzetessége, hogy viszonylag szerény →perifériával rendelkezik. Alapkiépítése rendszerint a →billentyűzettel egy tokba szerelt →központi egységből áll, kimeneti eszköze pedig általában a közönséges tévékészülék. →Háttértárolóként esetleg a hagyományos kazettás magnetofon csatlakoztatható.

HD (High Density) – nagyfokú tárolási sűrűséget biztosító →hajlékonylemez típusjele (→tárolási sűrűség).

helycserés rendezés (interchange sorting) – →operatív memóriában megvalósítható →rendezési módszer, amelynek lényege, hogy a rendezendő halmaz elemeit párosával összehasonlítják, s ha azok nem tesznek eleget a rendezési feltételeknek, helyüket felcserélik. Az eljárást addig ismétlik, amíg nem található egyetlen ilyen páros sem. Legegyszerűbb formája a buborékrendezés, ahol mindig a szomszédos elemeket cserélik fel.

helyesírás-ellenőrzés (spell checking) – modern →szövegszerkesztők speciális szolgáltatása, amely a szerkesztett szöveg ellenőrzését jelenti a megfelelő (természetes) nyelv helyesírási szabályainak alapján.

helyettesítési szabály (derivation rule) – a →szintaktikai elemzésben használt jelsorozat-átalakító szabály. Formája:

$$X \rightarrow Y$$

ahol X és Y olyan szövegek, amelyeknek szimbólumai →terminális és →nemterminális szimbólumok, a nyíl pedig azt jelenti, hogy az X szöveg a mondat származtatásakor helyettesíthető az Y szöveggel. Pl. ha a ~:

$$aab \rightarrow bCbb$$

akkor a W=aaabbc szövegből a következő új szöveg (szimbólumsorozat) képezhető:

$$W = aabCbbc$$

Levezetési (generálási) lépésnél az aktuális mondatformából úgy kapnak másikat, hogy a →nyelvtan valamely ~a bal oldalán szereplő szimbólumsorozatot a mondatformában a szabály jobb oldalán szereplő szimbólumsorozattal helyettesítik (→felülről lefelé haladó szintaktikai elemzés).

Redukciós lépésnél az aktuális mondatformából úgy kapnak újat, hogy valamely ~ jobb oldalán szereplő szimbólumsorozatot helyet-

tesítik a mondatformában a bal oldali sorozattal (→alulról felfelé haladó szintaktikai elemzés).

A – definíciója igen sokféle átalakítást tesz lehetővé, ezért a gyakorlatban bizonyos megszorításokat alkalmaznak. Noam Chomsky a megszorítások természete alapján négy nyelvtípust határozott meg (→Chomsky-hierarchia).

helyfüggetlen kód (position-independent code) – rendszerint kisszámítógépeknél használt programozási módszer. Olyan programok készítését jelenti, amelyek nem igényelnek címtranszformációt, bárhová töltik is az →operatív memóriába. Megvalósítása az olyan →relatív címek kizárólagos használatán alapul, ahol minden hivatkozás az aktuális →gépi utasításhoz van viszonyítva.

helyi hálózat (local area network) – korlátozott kiterjedésű (általában 1 km sugarú körzeten belüli) →adatátviteli hálózat, amelyben az →adatvégállomásokat és a feldolgozó számítógépeket fizikai adathordozók (huzalok, koaxiális kábelek) kötik össze. Alapvető jellemzői a nagy sebességű adatforgalom (1-100 Mbps), a →modemek hiánya és az alacsony hibaszázalékos adatátvitel.

helyi hiba (local error) – olyan →hiba, amely egy program →fordításának, →kapcsolásának v. végrehajtásának csak bizonyos részleteire van kihatással. A gyakorlatban fellépő hibák kezelésére rendszerint →hibaszinteket állapítanak meg, hogy így a felhasználóra bízhassák annak eldöntését, hogy egyes hibákat –knak v. →globális hibáknak tekint-e.

helymeghatározó mód (locate mode) – a →logikai [adat]állományok feldolgozásában használatos rekordelérési módszer. Lényege az, hogy az olvasó utasításra az →operációs rendszer közli a következő feldolgozandó rekord →címét a →puffermemóriában, de azt fizikailag nem mozgatja (nem tölti a megfelelő programváltozó területére), ill. író utasításnál a rendszer közli a következő szabad címet a puffermemóriában, ahová a programozónak kell elhelyeznie a következő kiírandó rekordot. A – abban különbözik a →mozgató módtól, hogy itt a felhasználóra hárul a logikai rekord esetleges fizikai mozgatása a rekord számára fenntartott memóriazóna és a puffermemória között, esetleg a teljes feldolgozás a puffermemória keretein belül valósul meg.

helyőr (cursor) – képernyős →terminálóknál olyan (sokszor villogó fényerősségű) speciális karakter (alsó kötőjel, csillag, nyíl, pont stb.), amely a következő kiírandó szimbólum helyét mutatja, vagyis referenciapontot képvisel a képernyő mezejében. Egy-egy alfanumerikus karakter kiírása automatikusan a következő szabad helyre mozdítja a –t (balról jobbra egy pozícióval). Ha az aktuális sor betelt, a – automatikusan a következő sor első pozíciójára ugrik. A soremelést végző billentyűvel a –t, függetlenül pillanatnyi helyzetétől, a következő sor bal első pozíciójára lehet helyezni. A billentyűzet tartalmazhat olyan →funkcionális billentyűket is (→nyílbillentyűk), amelyek kiírás nélkül is mozgathatják a –t balra, jobbra, fel, ill. le (rendszerint egyetlen karakterpozícióval).

heterogén adatstruktúra (heterogeneous data structure) – olyan →adatstruktúra, amelyben a különböző adatelemek különböző attribútumokkal rendelkeznek. Jellegzetes képviselője a számítástechnikában az →[adat]állomány.

hétköznapi tudás (common sense knowledge) – olyan, nem elméleti jellegű →ismeretek összessége, amelyeket az emberek – a környezet megismerésével párhuzamosan – az anyanyelvvel együtt sajátítanak el. Lényegében a szavak jelentésében rejlő egyszerű térbeli, időbeni és oksági összefüggésekre vonatkozik. Tartalmi gazdagsága, osztályozhatatlansága miatt a mai →szakértőrendszerek még képtelenek ábrázolni, ill. feldolgozni (ellentétben a →tárgyköri szakértő tudásával).

hétréteges referenciamodell (seven-layer reference model) – az ISO által javasolt kommunikációs →protokollmodell, amelynek célja a különböző szabványokon nyugvó számítógéprendszerek összekapcsolhatósága. Az elméleti modell 7 egymásra épülő réteget különböztet meg:

1) Fizikai réteg: meghatározza a fizikai összeköttetések mechanikai, elektromos és funkcionális jellemzőit, az összeköttetések létrehozását, fenntartását és megszüntetését támogató előírásokat úgy, hogy transzparens adatátvitelt biztosítson a fizikai adathordozókon keresztül;

2) Adatkapcsolati réteg: biztosítja adatblokkok hibamentes, megbízható átvitelét két hálózati csomópont között;

3) Hálózati réteg: megfelelő címzések, →adatútképzések révén biztosítja az adatblokkok veszteségmentes átvitelét a fogadó csomópontba.

Az első három réteg alapvető feladata tehát az információbitek átvitele a fizikai adathordozókon (fizikai réteg), az információbitek sorbarendezése, adatblokkokba építése (adatkapcsolati réteg), ill. az adatblokkok eljuttatása a fogadó csomópontokba (hálózati réteg). Ezen a három alsó szinten alakultak ki a →nyilvános adathálózatok, ill. a →csomagkapcsolásos szabvány.

A következő négy réteg szabványosítása napjainkban még tart:

4) Átviteli réteg: az alsó három (átvitelorientált), ill. a felső három (alkalmazásorientált) réteg közötti átmenet megvalósítója;

5) Szessziós v. adatkapcsolati réteg: különböző alkalmazások közötti párbeszéd biztosítása;

6) Adatbemutató réteg: az alkalmazói szintű adatcsere és adatértelmezés szabványos megvalósítója;

7) Alkalmazói réteg: feladata a külvilág és a kommunikációs hálózat közötti kapcsolat biztosítása.

heurisztika (heuristics) – bizonyos típusú feladatoknak olyan megoldási módszere, amely a gyakorlatból, az ismételt próbálkozásokból eredő tapasztalatokon alapszik, v. találékonyságra, ill. intuícióra támaszkodik. Nem garantálja a biztos megoldást, de tapasztalat szerint általában elég jó, s főleg elfogadható időn belül elérhető megoldáshoz vezet.

heurisztikus adatútképzés (heuristic routing) – olyan →adatútképzési eljárás, amelynél az adott forráscsomópontból eredő, különböző összeköttetésekből jövő szabványos →adatcsomagok által létrehozott késleltetést úgy használják fel a kimeneti adatcsomagok irányítására, hogy az illető csomópont vonatkozásában a legjobb összeköttetést nyerjék.

heurisztikus program (heuristic program) – minden olyan program, amely nem determinisztikus algoritmusokat valósít meg, hanem próbálkozás-hiba módszerrel működik. Tanulási elemeket is tartalmazhat, vagyis a válaszok alapján megváltoztathatja működését.

heurisztikus programozás (heuristic programming) – olyan →programozás, amelynél nem matematikailag v. logikailag megalapozott, hanem gyakorlati megfigyelésekből eredő, esetleg intuícióra támaszkodó módszereket használnak. Az esetleges megoldáshoz sokszor csak többszöri próbálkozás, az előző változatokból leszűrt tanulságok vezetnek.

hexadecimális ábrázolás (hexadecimal representation) – számoknak a 16 hatványai szerint rendezett ábrázolása. Az együtthatók jelkészlete: (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F). Minthogy $16=2^4$, négy bináris számjegy fogható össze egyetlen hexadecimális számjeggyé. A →bájtszervezésű számítógépekben a →bájt 8 bitpozícióján ábrázolt bináris információ kétjegyű hexadecimális számmal kényelmesebben kifejezhető.

hiba (error) – általános technikai értelmezésben: valamely művelet célja és eredménye között adódó különbség. Numerikus vonatkozásban az abszolút ~ a számítandó mennyiség valódi és számított értéke közötti különbség, a relatív ~ pedig az abszolút ~ és a számított érték hányadosa. A ~ értékét általában nem lehet előre pontosan meghatározni. →Hibabecslés segítségével korlátokat, hibaképleteket adhatnak meg, v. statisztikai meggondolásokat alkalmazhatnak a ~ legvalószínűbb értékének meghatározására. Számítástechnikai szempontból sokszor szükség van a kiindulási adatok ~inak kiegyenlítésére. Programozástechnikai vonatkozásban különleges jelentősége van a →szintaktikai és a →szemantikai hibának. Hibás algoritmus alkalmazása v. géphiba is ~t, értelmezési v. végrehajtási zavart okozhat a számítógépben.

hibabecslés (error estimation) – különböző forrásokból származó hibák hatásának felmérése a számítás eredményére. A ~ alapján hibaképletek, hibakorlátok adhatók meg arra vonatkozóan, hogy mennyire pontos a számított mennyiség.

hibacsomó (error burst) – →adatátviteli hálózatokban fellépő hibák olyan típusa, amikor az első és utolsó meghibásodott csatornaszimbólum között elhelyezkedő szimbólu-

mok meghibásodási valószínűsége sokkal nagyobb, mint az ezeken kívül eső szimbólumoké.

hibaelemzés (error analysis) – valamely számítási eljárás végeredményének közelítő →hibájára vonatkozó előzetes felmérés. Lehet progresszív, amikor a számítási eljárás előrehaladásával párhuzamosan veszik figyelembe a felmerülő hibákat, v. regresszív, amikor a számítási eljárás befejezése után a felgyűlt hibákat egyenértékű kumulatív hibával helyettesítik. Jelentősége az, hogy egységesen lehet tárgyalni a különböző típusú hibákat. A ~ nem veszi figyelembe a géphibákból v. a téves számításokból eredő hibákat.

hibafelderítő program (debugging program) – olyan speciális segédprogram, amelynek meghatározott moduljait a →kapcsolóprogram segítségével egy fejlesztés alatt lévő programhoz lehet kapcsolni, s ezáltal megvalósítható az →ellenőrzött végrehajtás v. a →lépésenkénti végrehajtás. A ~ jellege és funkciói a feldolgozási módtól függően változnak. →Kötegelt feldolgozásnál a program végrehajtása után adja meg a standard →kimeneti egységen a kívánt információkat. →Párbeszédes üzemmódban dolgozó rendszereknél lehetőséget ad a program végrehajtásának interaktív figyelésére, helyi beavatkozásokra, újrakezdésekre stb. A ~ moduljai rendszerint standard →modulkönyvtárakból építhetők be a felhasználói programokba.

hibajavító kód (error correcting code) – olyan →redundáns kód, amelynek redundanciája lehetővé teszi a vevő állomáson a hiba felismerését és a helyes kód rekonstruálását. Gyakorlatilag speciális hibajelző biteknek az információhordozó bitekhez való csatlakoztatásával valósítják meg, s így hibás adatátvitelnél különleges bitkombinációk jelennek meg. A nem javítható hibák jelzésére is alkalmas.

hibajelzés (error indication) – 1. A számítógéprendszer →központi egységében v. a →perifériában észlelt rendellenes állapotnak a gépkezelő (v. felhasználó) tudomására hozása. Végrehajtása v. a →vezérlőpulton elhelyezkedő optikai kijelzők segítségével (pl. memóriahibánál, paritáshibánál stb.) v. pedig a →konzolírógépen megjelenő szöveges információk útján történik (rendszerint a periféria működésében fellépett, az →operációs rendszer által felismert hibáknál). – 2. Különböző programok (→fordítóprogramok, →értelmezőprogramok, →kapcsolóprogramok, →betöltőprogramok, →könyvtárkezelő programok, →rendezőprogramok, felhasználói programok stb.) által a standard →kimeneti egységen v. az →interaktív terminálon közölt (szöveges v. kódolt) üzenet a feldolgozás közben észlelt logikai v. szervezési rendellenességekről.

hibajelző bit (error check bit) – a számítógép működésének v. a →kódolás helyességének ellenőrzésére – az információhordozó bitek mellé – rendelt redundáns bit.

hibajelző kód (error detecting code) – olyan →redundáns kód, amelynek redundanciája lehetővé teszi a vevő állomáson az adatátvitelben bekövetkezett hibák felismerését, de azoknak kijavítását nem (→hibajavító kód). Tkp. a legegyszerűbb →hibakorlátozási módszer az adatátvitelben. A gyakorlatban a →paritás ellenőrzése és az aránykódok használata terjedt el.

hibaképlet (error formula) – valamilyen numerikus eljárás pontosságára, vagyis a számított és a tényleges érték eltérésére megadható képlet.

hibák közötti átlagos idő (mean time between failures) – számítógéprendszer működésében két hibaleállás közötti átlagos idő, az illető rendszer működési biztonságát jellemző mutató. →Kötegelt feldolgozással működő rendszereknél nagyságrendje néhány száz óra, folyamatvezérlő rendszereknél viszont sokszor néhány ezer órát is el kell hogy érjen.

hibakorlát (limit of error) – a számítás pontosságára, azaz a számítás közben elkövetett (rendszerint eleve elkerülhetetlen) hiba nagyságára vonatkozó – általában becsült – érték.

hibakorlátozás (error limiting) – az adatátvitel megbízhatóságának növelésére alkalmazott módszerek összessége. Mivel az adatátvitel megbízhatósága általában 2-4 nagyságrenddel kisebb, mint a számítástechnikai elemek megbízhatósága, nagyfokú ~ra van szükség. A gyakorlatban az adatátviteli műveletek megbízhatóságát az →üzenetek méretének, a →hibakorlátozó kódolásnak és a →hibakorlátozó stratégiáknak megfelelő, összehangolt alkalmazásával fokozzák.

hibakorlátozó kódolás (error limiting encoding) – olyan →kódolás, amelynél a kódolt adatelemek célszerűen választott sorozatát tudatosan és meghatározott törvényszerűség szerint redundáns ellenőrző elemekkel látják el. Azokat a kódokat, amelyek csak a hibák felismerésére alkalmasak, →hibajelző kódoknak nevezik, ha viszont a meghibásodott kódszavakból bizonyos fajta hibák helyére is egyértelműen következtetni lehet, akkor →hibajavító kódokról beszélnek.

hibakorlátozó stratégia (error limiting strategy) – az adatátvitel megbízhatóságát növelni hivatott eljárás, amelyet a →hibakorlátozó kódolással összhangban használnak. Az adatátviteli rendszerekben általában a következő ~kat szokták alkalmazni: →visszhangellenőrzés, →automatikus ismétléskérés, →közvetlen hibajavítás.

hibalista (error list) – egy →rendszerprogram (→fordítóprogram, →kapcsolóprogram, →betöltőprogram stb.) által felfedezett hibáknak a standard →kimeneti egységen (rendszerint →nyomtatón) kiadott listája →kötegelt feldolgozásnál. →Párbeszédes üzemmódban ugyanis minden felfedezett hibát azonnal közölnek a →terminálnál ülő felhasználóval. A ~ban található szöveges információk v. hibakódok a hibák szintjére, természetére, a hibásnak talált forrásprogramsor számára, esetleg a feldolgozás fázisára utalnak. Egyes →operációs rendszerek a hibakiadást úgy valósítják meg, hogy a ~ tkp. egy speciális →[adat]állomány, amelyet a felhasználó utólag listázhat.

hibanapló (failure journal) – →operációs rendszerekben általában megtalálható olyan speciális →[adat]állomány, amelybe a rendszer maga helyez el információkat bizonyos nem végzetes, de rendellenes állapotokról. Ezeket az adatokat később fel lehet használni az egész rendszer működésének kiértékelésére, a →megelőző karbantartás megvalósítására. A személyi számítógépeknél ilyen szolgáltatásokat egyes programok és hálózati rendszerek tudnak adni.

hibaszint (error level) – a számítógépes feladatmegoldás különböző fázisaiban előfordulható hibák jellemzésére használatos fogalom. Az adott fázisban felmerülő hiba ~jének megfelelően a további fázisokat fel lehet függeszteni, egyes fázisokat át lehet ugrani (→kötegelt feldolgozásnál), v. esetleg a hiba figyelembevételével lehet folytatni a további feldolgozást. Az egyes ~eknek megfelelő döntések rendszerint implicit módon vannak meghatározva (a rendszerre jellemző módon), a legtöbb →operációs rendszer megengedi viszont azt, hogy explicit deklarációkkal maguk a felhasználók döntsenek a hibák utáni tevékenységekről.

hiba utáni visszaállítás (error recovery) – programozástechnikai értelemben olyan módszer, amely lehetővé teszi a programok végrehajtásának folytatását bizonyos hibák bekövetkezése után is. Felhasználói szinten ez olyan →szubrutinok beépítését jelenti a programba, amelyek aritmetikai hibák (nullával való osztás, →túlcsordulás stb.) v. az →[adat]állományok kezelésében észlelt hibák megfelelő kezelését elvégzik, és a helyzetnek megfelelő döntésekkel folytatják v. esetleg felfüggesztik a program végrehajtását. Erre egyes magas szintű →programozási nyelvek (pl. →PL/1, →COBOL, →Ada stb.) →forrásnyelvi szintű lehetőségeket is biztosítanak, másoknál →asszembler nyelvű szubrutinok beépítése szükséges.

Különleges jelentősége van a ~nak a →fordítóprogramok előállításában. A fordítás közben észlelt →szintaktikai hibák felismerése és kijelzése után igen fontos a forrásprogram egy olyan pontjának a felismerése, ahonnan az analízis folytatható, máskülönben az az abszurd helyzet állna elő, hogy egy fordítás egyetlen hiba felismerését tenné lehetővé.

hibaüzenet (error message) – valamely program →fordításával, →kapcsolásával, →betöltésével v. végrehajtásával kapcsolatos hibás állapotokra vonatkozó szöveges v. kódolt információ, amelyet a program v. az →operációs rendszer közöl a felhasználóval v. a gépkezelővel a standard →kimeneti egységen, a →konzolírógépen v. az →interaktív terminálon.

hibrid számítógép (hybrid computer) – →analóg számítógépből és (rendszerint közepes v. kis kapacitású) →digitális számítógépből álló rendszer, amelyben a kapcsolatot →analóg-digitális, ill. →digitális-analóg konverterek teremtik meg. Kihasználja a két különböző jellegű számítási rendszer előnyeit (a

differenciálegyenletek gyors integrálási lehetősége az analóg gépen, ill. a nagyfokú pontosság, tárolási lehetőség stb. a digitális számítógépen). A ~ előnyösen használható bizonyos dinamikus rendszerek szimulálásánál, vezérlésénél és optimalizálásánál.

hidegindítás (cold start) – →operációs rendszer munkába állítása a számítógép bekapcsoláa után. Rendszerint rendszerbetöltést és →inicializálást foglal magában. ~t tesz kötelezővé általában a →rendszerösszeomlás is (→melegindítás).

hierarchikusan strukturált adatok (hierarchical structured data) – a →rekurzivitás megnyilvánulása az adatszerkezetekben. Olyan komplex adatszerkezetet jelent, amelyben a felépítő szerkezetelemek azonos típusúak a föléjük rendelt adatszerkezettel. Pl. olyan →lista, amelynek elemei egyszerű adatelemek és/vagy allisták, az allisták elemei pedig szintén egyszerű adatelemek és/vagy allisták stb. A ~at rendszerint láncoltan ábrázolják az →operatív memóriában.

hívás (call) – programozástechnikai értelemben valamely önálló programegység aktivizálása, végrehajtásának kezdeményezése (→eljáráshívás, →makrohívás). Tkp. két alapvető tevékenységből tevődik össze: a programrész működéséhez szükséges adatok előállítása és átadása (paraméterátadás), ill. a vezérlés átadása az aktivizálandó programegységre.

hívási sor (calling sequence) – az →eljáráshívást előkészítő és végrehajtó →gépi utasítások speciális sorozata. Feladatai: az →aktuális paraméterek előállítása, a visszatérési cím előkészítése, valamint a vezérlés átadása a hívott eljárásnak. Magas szintű programozási nyelveknél a ~ előállítása a →fordítóprogram feladata, →asszembler nyelven viszont a programozónak kell gondoskodnia róla.

hivatkozás (reference) – valamely programozási objektumhoz történő hozzáférés az illető objektum szimbolikus nevének (→azonosítójának) segítségével. A ~ lokális, ha a hívással azonos programrészben definiált objektumra vonatkozik, és külső (globális), ha a hivatkozott objektumot más programegységben definiálták. A külső ~ feloldása a →fordítóprogam (→asszembler), a →kapcsolóprogram v. a →betöltőprogram feladata.

hivatkozási nyelv (query language) – az →adatbázisoknál használt →adatleíró, ill. →adatkezelő nyelvek kiindulási (magas szintű) nyelve, amelybe beépülnek a speciális funkciókat megvalósító struktúrák. Általánosan használt ~ a →COBOL, a →PL/1 stb.

holtpont (deadlock) – →végleges kizárás

homogén adatstruktúra (homogeneous data structure) – olyan →adatstruktúra, amelyben az összes adatelem azonos attribútumokkal rendelkezik. Jellegzetes képviselője a →tömb.

„honos" számítógép (host computer) – →gazdaszámítógép

hordozhatóság (portability) – programozástechnikai szempontból: programtermékeknek az a tulajdonsága, hogy új környezethez könnyebb módosítani őket, mint újraírni. Ez a tulajdonság igen fontos, ugyanis a felhasználóknak az az érdekük, hogy egy adott számítógépen kifejlesztett programtermék minél kisebb változtatásokkal futtatható legyen más számítógépeken is. A ~ megvalósítására különböző módszereket dolgoztak ki, ezek legrégibb képviselője a →magas szintű programozási nyelv. Az ilyen nyelven kódolt programok elméletileg minden olyan gépen végrehajthatók, ahol létezik az adott nyelv →fordítóprogramja. Ez a módszer sokat nyert hatásosságában a programozási nyelvek szabványosításával, vagyis szigorú szabályok szerinti definíciójával, a fordítóprogramok gyakorlati megvalósításaiban azonban még mindig igen sok eltérés mutatkozik. A módszer finomításaként merült fel az a javaslat, hogy használjanak egy univerzális belső nyelvet a fordítóprogramok írására, s ezt a nyelvet implementálják minden számítógépen. Mivel ezt a nyelvet →forrásnyelvként, →tárgynyelvként és →fordítóprogram-író nyelvként is kellene használni, a vele szemben támasztott követelmények bizonyos fokig ellentmondanak egymásnak, így a módszer nem terjedt el a gyakorlatban. Érdemleges eredményt az →UNIX →operációs rendszer →C nyelve, ill. az egyre nagyobb érdeklődésnek örvendő →FORTH nyelv képvisel (ez utóbbi főleg a →mikroprocesszoros fejlesztésekben).

hordozható számítógép (laptop computer) – saját energiaforrással (akkumulátor,

elemek) rendelkező, tehát tápházózattól függetlenül üzemeltethető →mikroszámítógép.

hozzáférés (access) – 1. Az a művelet, amelynek révén az →operatív memória meghatározott rekesze elérhetővé válik a →központi vezérlőegység számára, hogy oda megfelelő adatot rögzítsen v. a rekesz tartalmát kiolvassa. A →címezhetőség elve alapján valósul meg. A →címszámítással meghatározott →effektív cím, az →abszolút cím v. az →asszociatív címzés az operatív memória bármelyik rekeszét közvetlenül hozzáférhetővé teszi a központi vezérlőegység számára. →Multiprogramozásos környezetben a ~ bizonyos védelmi előírásoknak is meg kell hogy feleljen (→hozzáférési kulcs). – 2. Az a művelet, amelynek révén az →adathordozón adat rögzíthető, ill. onnan visszanyerhető. Az adathordozó típusától, az adathordozó és a tárolandó információk megszervezésétől függően különböző típusú →hozzáférési módszerek léteznek: szekvenciális, indexszekvenciális, közvetlen ~. – 3. Multiprogramozásos környezetben az egyes felhasználók lehetősége, hogy a számítógéprendszer bizonyos →erőforrásait használják (→hozzáférési szint).

hozzáférési kulcs (access key) – bináris vektor, amely tartalmán keresztül lehetővé teszi v. megtiltja az →operatív memória adott zónájához való →hozzáférést. Használata biztosítja a különböző, az operatív memóriában egyidejűleg jelenlévő programokhoz rendelt memóriazónák kölcsönös védelmét s ezáltal →multiprogramozás megvalósítását (→programállapotszó).

hozzáférési módszer (access method) – →[adat]állományokba szervezett információk tárolására és visszakeresésére használható módszer, amelynek háromféle alapváltozata ismeretes: →szekvenciális, →indexszekvenciális és →közvetlen ~. Az →adathordozó típusa alapvetően befolyásolja a felhasználható ~t. Pl. a →lyukkártya, a →mágnesszalag stb. természetükből adódóan csakis a szekvenciális ~t teszik lehetővé, míg a →mágneslemez – címezhetősége révén – mindhárom alaptípust használhatja.

A ~t meghatározó másik szempont az [adat]állomány szervezési típusa. →Szekvenciális [adat]állomány (még ha mágneslemezen is van tárolva) csakis szekvenciális ~t engedélyez. A →direkt szervezésű [adat]állomány szintén csak egyik fajta, a direkt ~t teszi lehetővé. Ezzel szemben →indexszekvenciális [adat]állomány előállításakor csak szekvenciális ~, feldolgozásnál pedig bármilyen ~ használható.

hozzáférési modul (access module) – az →operációs rendszer olyan alapvető →szubrutinja, amely a →logikai [adat]állományokkal kapcsolatos műveleteket végzi el a felhasználó utasításai alapján. A →magas szintű programozási nyelvek megfelelő utasításokkal (READ, WRITE, FIND stb.), az →asszembler nyelvek pedig →makroutasításokkal rendelkeznek a ~ok aktivizálására.

hozzáférési szint (access level) – több felhasználóval dolgozó →operációs rendszerek védelmi mechanizmusának jellemzője. Célja a különböző programok közötti nemkívánatos kölcsönhatások megakadályozása. A programok természete és fontossága alapján az →operatív memóriában egyszerre jelenlévő minden programhoz egy-egy ~et utalnak, ez egyben az →erőforrásokkal szemben támasztott igények jogosságát is jellemzi.

hozzáférési zár (privacy lock) – →adatbázisok védelmi mechanizmusának fontos eleme: a jogtalan hozzáférést megakadályozó olyan másodlagos adat, amelynek tartalmát a felhasználónak ismernie kell ahhoz, hogy az adatokhoz hozzáférhessen.

hozzárendelés (assign) – valamely →erőforrás (rendszerint →periferikus mű) rendelkezésre bocsátása. Ez a kiosztás általában v. az egész →munka, v. annak egyik fázisa idejére történik.

Huffmann-kód (Huffmann-code) – az →adattömörítés megvalósítására használható, változó hosszúságú karakterkód. Alapelve az az észrevétel, hogy ha egy bizonyos feldolgozástípusnál az egyes karakterek eloszlása erősen aszimmetrikus, akkor a leggyakrabban használt karaktereket rövid, a ritkábban előfordulókat pedig hosszabb kódszavakkal ábrázolhatják. Így a leggyakrabban előforduló karaktert 1 bittel, a következőket 2-vel stb. lehet ábrázolni. Minél egyenlőtlenebb a karakterek eloszlása, annál hatékonyabb a ~. Pl. angol szövegekben az ábécé betűi váltakozó gyakorisággal fordulnak elő, s a ~ karakterenkénti átlagos bitszámát 4,12-re lehet

leszorítani. A legtöbb kereskedelmi-gazdasági adatfeldolgozásban az eloszlás ennél jóval egyenlőtlenebb, s így gyakran elérhető a karakterenkénti átlagos 3 bites kódolás is.

hulladékgyűjtés (garbage collection) – →dinamikus memóriakiosztásban használt stratégia. Lényege, hogy a pillanatnyilag felszabaduló memóriazónákat összegyűjtik, majd dinamikusan, a felmerült igényeknek megfelelően újra átcsoportosítják.

I, Í

időosztásos multiplexelés (time-division multiplexing) – →multiplexelési módszer: lényege, hogy a nagyon gyors →multiplexcsatorna meghatározott időintervallumokra különböző csatornákhoz van hozzárendelve. A kiosztás megismételhető rendszeresen (rögzített ciklus), és megtehető a pillanatnyi igényeknek megfelelően is (dinamikus kiosztás).

időosztásos rendszer (time-division system) – többfelhasználós rendszerek egyik alapvető megvalósítási típusa. Olyan használati üzemmódban működő számítógépes rendszer, amely lehetővé teszi, hogy a rendszerhez kapcsolt bármelyik felhasználó bármelyik pillanatban látszólag a teljes erőforrás-készlettel rendelkezzék. Az – lényege hardverszempontból az, hogy az →operációs rendszer egy belső óra működése és speciális megszakító utasítások révén a gépi erőforrásokat (elsősorban a →központi vezérlőegység idejét) a párhuzamosan futó programok között egységnyi időszeletek tartamára osztja ki. A periferikus egységek és a központi egység sebessége közötti több nagyságrendű különbség kiegyenlítésével az – azt a látszatot képes kelteni, hogy minden felhasználó saját rendszerrel rendelkezik. Az – szempontjából a teljes számítógépes munkaválaszték (programok írása, próbája, javítása, végrehajtása) →párbeszédes üzemmódban, az →[adat]állományok létesítése, törlése, módosítása, lekérdezése stb. a felhasználás valós idejében történik.

időszelet (time slice) – az az időtartam, amit az időosztásos →operációs rendszer összefüggően egy felhasználó rendelkezésére bocsát.

igazságérték (truth value) – →logikai érték

igazságtáblázat (truth table) – egy- v. többváltozós →logikai függvény értékeit az argumentumok függvényében feltüntető táblázat. A logikai tételek bizonyításának hasznos segédeszköze.

ige (verb) – a →COBOL programozási nyelvhez tartozó utasítások első →kulcsszavaként használt →lefoglalt szó, amely meghatározza az utasításhoz tartozó általános tevékenységet (pl. ADD, MOVE, TRANSFORM stb.).

igényre szabott információs rendszer (tailored information system) – olyan információs rendszer, amely az adatfeldolgozást nem általánosított, hanem a konkrét feladat jellege (az adatszerkezetek és adatkezelési igények) által meghatározott eszközökkel végzi. Költségesebb megoldás, mint az →általánosított információs rendszernek az adott konkrét körülményekre vonatkoztatott felhasználása, de sokkal jobb hatásfokkal képes kielégíteni az egyéni felhasználó konkrét igényeit.

ikon (icon) – képi szimbólum fogalmak, műveletek stb. azonosítására. Elsősorban egyes →menüvezérelt üzemmódban dolgozó rendszerekben használják, különösen azokban a grafikus rendszerekben, ahol a műveletválaszték igen gazdag, s a természetes nyelvű →menü v. elfoglalná az egész képernyőt, v. →görgetni kéne egy viszonylag kisebb →ablakban. (Pl. különböző grafikus mintázatokkal történő térkitöltésnél.)

illegális karakter (illegal character) – jól meghatározott →karakterkészlettel rendelkező →automata, →programozási nyelv stb. szempontjából minden olyan karakter, amely nem tartozik az illető karakterkészlethez. –ként kezel egy-egy programtermék (pl. →fordítóprogram, →adatbázis-kezelő rendszer stb.) minden olyan karaktert is, amely hozzátartozik ugyan a karakterkészlethez, de használata tiltott pl. →azonosítókban, →[adat]állománynévben stb.

illegális utasítás (illegal instruction) – olyan →utasítás, amely nem tartozik hozzá egy konkrét számítógép →utasításkészletéhez. Megjelenése egy végrehajtás alatt álló →programban →csapdát jelent, s a végrehajtás felfüggesztődik. Ezt a jelenséget szokták használni a különböző szoftver →erőforrások aktivizálására (→kiterjesztett gép, →felügyelőprogram-hívó utasítás). –ként kezel a →felügyelőprogram minden →privilegizált utasítást is, ha az felhasználói programban jelenik meg.

illesztőegység (interface) – két, különböző funkcionális jellemzőkkel rendelkező berendezés (esetleg programtermék) között kapcsolatot teremtő →hardver- v. →szoftveregység. Magában foglalja a csatlakozási pontok fizikai paramétereit, a rajtuk áthaladó

jelek értelmezését és időzítését, a működési sorrendet stb.

implementációfüggőség (implementation dependency) – valamely →programozási nyelv olyan részeinek jellemzője, amelyek egyes →fordítóprogramoknál, ill. →értelmezőprogramoknál különbözhetnek a megvalósítás módjában és értelmezésében, sőt teljesen hiányozhatnak is. Ilyen részek lehetnek pl. az →[adat]típusok belső ábrázolása v. az →[adat]állományokra vonatkozó szervezési és feldolgozási módszerek. Az ~ nemkívánatos jelenség, ezért igyekeztek minimálisra csökkenteni. Egyik ilyen irányzat a programozási nyelvek szabványosítása.

implicit címzés (implied addressing) – elsősorban →gépi nyelvekben használható olyan címzés, amelynél az →utasításban nem kell feltüntetni az →operandus címét, ugyanis azt egyértelműen meghatározza az →utasításkód. Így pl. ~t lehet használni az aritmetikai műveletekben, ahol az első operandus, ill. az eredmény címe implicite az →akkumulátor. Ugyancsak ~t használnak a →veremműveletek (PUSH, POP), amelyeknek implicit címe mindig a →verem csúcsa.

implikáció (implication) – kétargumentumú →logikai függvény, amely akkor és csakis akkor hamis, ha az első argumentum igaz, a második viszont hamis. Igazságtáblázata a következő:

→	0	1
0	1	1
1	0	1

index[1] (subscript) – 1. A →tömb (mint homogén adatstruktúra) elemeinek tárolására, ill. visszakeresésére szolgáló mechanizmus. A tömb elemeit az →operatív memóriában ~-ük szerint növekvő sorrendben helyezik el. Az ~-ek száma megadja a tömb dimenzióját. – 2. Magas szintű →programozási nyelvekben a →minősítés eszköze. A tömbadatszerkezet minden eleme ugyanazt a szimbólumot kapja névnek (→azonosítónak), tehát szükség van egy olyan →minősítőre, amely lehetővé teszi az egyes elemekre mint önálló adatokra való hivatkozást. A tömb elemeinek minősített neve a tömb azonosítójából és az azt követő ~-listából áll. Az ~-listát kerek zárójelek közé zárt, egymástól vesszővel elválasztott ~-ek alkotják (pl. a →FORTRAN, a →COBOL, a →PL/1 stb. nyelvekben), az →ALGOL, a →Pascal stb. nyelvek szögletes zárójeleket használnak. Az ~ általában egész szám v. egész [adat]típusú adat, néhány FORTRAN és más implementációban viszont kibővítették az ~-ek →szintaxisát azáltal, hogy ~-ként megengednek aritmetikai kifejezéseket v. ezek bizonyos alosztályait. A programozási nyelvet megvalósító →fordítóprogram feladata, hogy az ~(-ek) alapján olyan →gépi utasítássort generáljon, amely kiszámítja az adott tömbelemeknek a tömb kezdetéhez viszonyított relatív címét. →Erősen típusos nyelvekben az ~ bármilyen →sorszámozott [adat]típus lehet.

index[2] (index) – 1. →Gépi nyelvekben az →indexelt címzés megvalósításának eszköze, az →indexregiszter tartalma. Lényegében az utasítás →címrészében található →eltoláshoz viszonyított →relatív cím kifejezésére szolgál. – 2. Olyan másodlagos adat, amely meghatározza egy adatelem →[adat]állományon belüli logikai (logikai ~) v. fizikai (fizikai ~) helyét, azaz címét (→indexszekvenciális [adat]állomány). Az ~ tkp. az a legnagyobb (v. legkisebb) →rendezési kulcs, amely azonosítja azt az →adathordozói területet, ahol már csak szekvenciális hozzáférés lehetséges (→sávindex, →cilinderindex, →főindex).

indexállomány (index file) – →adatbáziskezelő rendszerekben használt speciális →[adat]állomány, amely az adott →szekvenciális [adat]állomány logikailag rendezett sorrendjét meghatározó →indexeket tartalmazza. Egyetlen [adat]állományhoz több ~ is rendelhető, ezek lehetővé teszik ugyanannak az [adat]állománynak több kritérium szerinti rendezett feldolgozását is, anélkül hogy az illető [adat]állományt fizikailag is át kellene rendezni.

indexelt cím (indexed address) – olyan →effektív cím, amelynek kiszámítása az utasítás címrészében található →eltolás és az →indexregiszter tartalmának összegezése útján történik. Az ~zést →indirekt címzéssel összekapcsolva is lehet használni. A címszámítás sorrendjét tekintve a vegyes címzés →előindexeléssel v. →utóindexeléssel valósulhat meg.

indexes változó (subscripted variable) – magas szintű →programozási nyelvekben a

→tömbökkel kapcsolatos fogalom: arra vonatkozik, hogy a tömb →azonosítójával tkp. a tömb elemeire mint önálló →változóra lehet hivatkozni.

indexregiszter (index register) – →indexelt címzésnél az →index értékét tartalmazó →regiszter. Tartalmát az utasítás végrehajtása előtt a címrészben szereplő →eltoláshoz adják, így alakítva ki az operandus →effektív címét. Egyes architektúrákban az ~ek speciális regiszterek, vagyis csak erre a műveletre használhatók, s az indexrész változtatását automatikusan végezhetik. Más rendszerekben az ~ek szerepét az →általános regiszterek látják el, s ilyenkor tartalmuk változtatása a felhasználói program feladata.

indexszekvenciális [adat]állomány (indexed-sequential file) – olyan →logikai [adat]állomány, amelyet címezhető →adathordozón (rendszerint →mágneslemezen) szekvenciálisan állítanak elő, de feldolgozása történhet mind szekvenciálisan, mind közvetlen módon. Az ~ előállításakor a →rekordok egy v. több →rendezési kulcs szerint rendezve kerülnek az adathordozóra, blokkolt v. blokkolatlan formában (→blokkolás), miközben a közvetlen (direkt) elérést biztosító indextáblák is felépülnek. Az ~ három elkülöníthető részből épül fel. Az elsődleges adatterületre kerülnek az ~ rekordjai az [adat]állomány előállításakor. A túlcsordulási területre feldolgozás közben kerülnek új rekordok; ezeket blokkolatlan formában, láncolt szervezésben helyezik el. Az indexterületre kerülnek azok az indextáblázatok, amelyek lehetővé teszik a rekordok majdnem közvetlen elérését. Az indextáblázatok az ~ előállításakor épülnek fel, és az [adat]állomány egész élettartamára változatlanok maradnak. Rendszerint →sávindexeket és →cilinderindexeket tartalmaznak, nagy adathalmazoknál használhatnak esetleg egy harmadik szintet is, a →főindexet. Az [adat]állományban való keresés meggyorsítására az indextáblázatokat rendszerint az [adat]állomány megnyitásánál az →operatív memóriába töltik.

indexszekvenciális alap-adatelérési technika (basic indexed-sequential access method) – →indexszekvenciális [adat]állományok feldolgozására alkalmazott →alap-adatelérési technika.

indexszekvenciális hozzáférési módszer (indexed-sequential access method) – címezhető →adathordozón (rendszerint →mágneslemezen) tárolt rendezett adatok elérésére használt módszer. Az →indexszekvenciális [adat]állomány →rekordjait növekvő kulcs szerint rendezve rögzítik az adathordozón az [adat]állomány előállításakor. Ezzel párhuzamosan építik fel az indextáblázatokat is, amelyek a visszakeresés alapvető eszközei. Az indextáblázatok hierarchikusan felépített táblázatok, általában kétszintűek (→sávindex és →cilinderindex), nagyon nagy adathalmazoknál harmadik szintet is alkalmazhatnak (→főindex).

Az ~ tkp. a szekvenciális és a közvetlen hozzáférések összehangolása. Adott kulcscsal rendelkező adat visszakeresése két fázisból tevődik össze. Először az indextáblázatok segítségével meghatározzák azt a →cilindert és azt a →sávot, ahol az adat biztosan megtalálható. Az adathordozó címezhetősége folytán ez a keresés közvetlen hozzáférést biztosít. Második ütemben a meghatározott sávon szekvenciális kereséssel jutnak az igényelt rekordhoz.

indexszekvenciális sorbaállító alapadatelérési technika (queued indexed-sequential access method) – →indexszekvenciális [adat]állományok feldolgozására alkalmazott →sorbaállító adatelérési technika.

indextúlcsordulás (subscript overflow) – →tömbökkel dolgozó programok hibás működésének gyakori okozója. Azt jelenti, hogy az →index (esetleg indexkifejezés) alapján kiszámított →cím az adott tömb területén kívül esik. Az ~ csak a program végrehajtása közben ismerhető fel (sokszor igen nehezen, ugyanis hatása általában késleltetve jut kifejezésre, a hibásan felülírt memóriazónák miatt). Az ilyen típusú hibák kiszűrése a →fordítóprogramok, rendszerint a programozó kifejezett kívánságára, ellenőrző szubrutinokat építenek a →tárgyprogramokba, amelyek minden →indexes változóra való hivatkozás előtt megvizsgálják, hogy az indexkifejezés kiszámított értéke nagyobb-e, mint a tömb hossza. Ha igen, hibajelzés után a program további végrehajtását felfüggesztik.

indirekt cím (indirect address) – olyan →cím a →gépi utasításban, amely nem a tu-

lajdonképpeni operandust tartalmazó rekeszre mutat, hanem olyanra, ahol annak tényleges címét tárolták (nevezik még közvetett címnek is). Ez a címzési mód, a konkrét számítógéptől függően, több szintre is kiterjedhet, tehát legalább kettő, de több memóriaciklust is igényelhet. Az →indexelt címzéssel vegyesen is használható (→előindexelés, →utóindexelés).

infix forma (infix form) – aritmetikai kifejezések felírásának olyan formája, amelynél a műveleti jelet az operandusok közé helyezik (ez másküIönben a köznapi használati forma is). Pl. az A és B összeadásának ~ja :

$$A + B$$

(Más, a számítástechnikában sokkal jelentősebb forma a →postfix forma és a →prefix forma.) A műveletek elvégzésének sorrendjét az algebrából ismert prioritási szabályok vezérlik.

információ (information) – eredeti értelmezésben: olyan tájékoztatás, közlés, adat, ismeret, hír stb., amely a címzett által értelmezhető, számára tényleges v. potenciális értelme van, s amelynek célja a bizonytalanság csökkentése, a lehetséges alternatívák közötti választás elősegítése. Számítástechnikai, bővített értelmezése általában az ismeretre vonatkozik, amely új, addig általa nem ismert vonatkozásokat közöl az emberrel v. a számítógéprendszerrel a környező valóságról. A számítógéprendszerek →adatokkal dolgoznak, amelyek az ~ fizikai ábrázolásai, s csak ilyen értelemben mondható, hogy a számítógépek ~t dolgoznak fel. A gépi feldolgozás szempontjából csak az a fontos, hogy egyértelműen azonosítani lehessen az adatok eredetét. Nem szükséges ismerni az ~k jelentését, elegendő az adatok ismerete.

Az →információelmélet és az adatfeldolgozás egyesített értelmezése szerint az ~ egy hordozóból és a mellé rendelt jelentésből álló logikai egység.

információábrázolás (information representation) – Az →információ egy hordozó és a mellé rendelt jelentés logikai egysége. Az ~ az első összetevőre vonatkozik, s magában foglalja a hordozó közegét és az információ →kódolásának módját. A hordozó közege v. →adathordozó (→lyukszalag, →lyukkártya, →mágnesszalag, →mágneslemez stb.), v. a →kommunikációs csatornákon áthaladó impulzussorozat. Az ~ másik fontos feladata annak a kódolási formának a megválasztása, amelynek révén az adott információ minimális jelhalmazzal válik hordozhatóvá.

információelmélet (information theory) – a matematika →információkkal, azok létrehozásával, mérésével, átalakításával, tárolásával és továbbításával foglalkozó ága. Szoros kapcsolatban áll a →kibernetikával és a valószínűségelmélettel. Megalapozója C. E. Shannon, aki a →bit fogalmát az információtartalom és a döntési tartalom mértékegységeként bevezette. Az ~ fontos eredménye az információ mennyiségi meghatározása, az →információmennyiség bevezetése.

információhozam (information rate) – az →adatátviteli hálózatok adó-, ill. vevőállomásait jellemző ama →információmennyiség, amelyet egységnyi idő alatt leadhatnak, ill. fogadhatnak. Elfogadható mennyiségű adatátvitel megvalósításához az adó ~a sohasem haladhatja meg a →kommunikációs csatorna átviteli sebességét.

információküszöb (information threshold) – az az érték, amely meghatározza, hogy milyen intervallumban változtatható a bemeneti jel anélkül, hogy arra a kimenet reagálna.

információmennyiség (information's amount) – 1. Rögzítési, feldolgozási v. átviteli műveletben részt vevő olyan fizikai szimbólumok száma, amelyek a valósággal kapcsolatban →információt hordoznak. – 2. Az információforrás által kibocsátott szimbólumok átlagos számának és az entrópiának a szorzata. Az információelmélet alapvető tétele, hogy adott üzenet ~e az információ konkrét tartalmától és az átvitel formájától függetlenül mérhető, ha ismerik a lehetséges üzeneteket és ezek előfordulási valószínűségét.

információminőség (information quality) – az →információt a rendeltetésének való megfelelése alapján minősítő érték, vagyis annak mértéke, hogy a címzettnél mennyire csökkenti a bizonytalanságot, ill. mennyire segíti elő a különböző cselekvési lehetőségek között a választási lehetőséget.

információrejtés (information hiding) – modern →programozási nyelveknek az a tulajdonsága, hogy az adatstruktúrák, ill. az al-

goritmusok implementálási részleteit hozzáférhetetlenné teszik a felhasználók számára. Ennek köszönhetően elkerülhetők azok a hibák, amelyek e részletek esetleges félreértéséből v. helytelen használatából erednek, s szinte rákényszerítik a felhasználót, hogy kizárólagosan a megoldandó problémára jellemző struktúrákban gondolkozzék, függetlenül a programot majdan ténylegesen végrehajtó számítógéptől.

Információtelítettség (information saturation) – a kimenet és a bemenet arányának az a felső határa, amelyen túl a bemenet megváltoztatása többé nem idéz elő változást a kimeneten.

Információ-visszakeresés (information retrieval) – az adatfeldolgozás egyik jellegzetes területe, amely magában foglalja az adatok elemzését, rendszerezését, rögzítését és lokalizálását, és amelynek alapvető célja speciális információk nyerése tárolt adatokból. Az – céljára készült programtermékek feladata megfelelő információk szolgáltatása a felhasználók által megfogalmazott kérdésekre. Kidolgoztak olyan rendszereket, amelyekkel a nagy adattömegek úgy kezelhetők, hogy az →adatbankkal való együttműködés során (ahol az adatokat a legkülönbözőbb szempontok szerint tárolták) szelektív információkat lehet visszanyerni. Nagyon lényeges, hogy nemcsak általánosan, hanem kérdéskritériumok alkalmas kombinációjával is lehet a választartományt behatárolni. Az – jellegzetes alkalmazási területe a bibliográfiai nyilvántartás, az orvosi diagnosztika, a statisztikai előrejelzés stb.

Informatika (informatics) – olyan multidiszciplináris tevékenység, amely az információfeldolgozás mellett magában foglalja az átalakítás előkészítésére, elvégzésére és az eredmények közlésére szolgáló eszközöket is. A számítástechnika fejlődésével más területek is bekapcsolódtak az – hatáskörébe: az orvostudomány, a nyelvészet, a gazdasági ügyvitel, az adminisztráció, a tanügy, a jogtudomány, a művészet stb. Korszerű értelmezésben az – magában foglalja mindazokat a tevékenységeket, amelyek az ember fizikai és intellektuális teljesítményének növelésére szolgáló rendszerek tervezésére, megvalósítására és kihasználására irányulnak.

Inicializálás (initialization) – az az eljárás, amely programok (v. programegységek) végrehajtásának megkezdése előtt a működés szempontjából meghatározó jellegű →konstansok, →változók, →mutatók stb. állapotát az előírt kezdőértékekre állítja. Számos →fordítóprogram a lefordított kódhoz olyan részeket ad hozzá automatikusan, amelyek bizonyos változóknak értéket adnak, [adat]állományokat megnyitnak stb. Ilyen az MS-DOS-ban az AUTOEXEC.BAT végrehajtása is.

Nagygépeken különleges jelentősége van az →operációs rendszerek –ának, amit minden egyes munkanap kezdésekor, ill. mindannyiszor el kell végezni, valahányszor a rendszer bizonyos – a →rendszergenerálás-kor beállított – paramétereit meg akarják változtatni (pl. a →partíciók méretének és/vagy számának változtatása, a →periferikus művek szimbolikus címének v. hozzárendelhetőségének módosítása, a standard →rendszerállományok jellemzőinek újrabeállítása stb.).

Inkrementálás (incrementation) – a programozástechnikában olyan művelet, amely a memóriarekesz v. →regiszter tartalmának adott konstans értékkel való növelését jelenti anélkül, hogy ezért az ott található operandust le kéne hívni a memóriából. A konstans értéke rendszerint 1, de lehet más egész érték is. Tekintettel arra, hogy az →iterációs eljárások, →ciklusok lényegében –ra vezethetők vissza, a legtöbb számítógép →utasításkészletében megtalálható az –t végző utasítás. Egyes architektúrákban speciális címzési módszerek is használják az –t (→növekményes címzés).

Inkrementális fordítás (incremental compiling) – →párbeszédes üzemmódhoz alkalmazkodó →fordítás, amely lehetővé teszi a programírás, a →szintaktikai és →szemantikai analízis és a fordítás egyetlen menetben történő megvalósítását. A speciális szerkezetű →fordítóprogram utasításonként, programegységenként (inkrementumonként) fordít. A lefordított részeket utólag kapcsolják egységes programmá, v. esetleg csak az →értelmezés módszerével történő végrehajtás során állnak össze azzá. A felhasználó programjának sorait tetszés szerinti sorrendben közölheti a rendszerrel, a program végrehajtása a növekvő utasításszámokat és nem a közlés

sorrendjét fogja követni. Noha jelentős előnyei vannak (csupán egyes inkrementumok írása, javítása és újrafordítása, a programok lépésenkénti végrehajtása stb.), komoly hátránya, hogy a programok tördelése miatt egyes döntéseket a →kapcsolásig, másokat pedig a végrehajtásig is el kell halasztani (ugyanis adott pillanatban nem áll rendelkezésre a fordításhoz szükséges összes szemantikai attribútum). Éppen ezért az ~ módszerét elsősorban a programfejlesztés és tesztelés fázisában, párbeszédes üzemmódban használják, a végleges forrásprogramot pedig egy utolsó teljes fordítás útján alakítják végső →tárgyprogrammá.

integráció (integration) – számítástechnikai értelemben az adatfeldolgozás olyan megszervezése, hogy a felhasználói alrendszerek szoros kapcsolatban, ne pedig egymástól függetlenül működjenek. Funkcionális az ~, ha az önálló alrendszerek egységes rendszerbe történő összevonása az egyes funkciókat megvalósító algoritmusok összehangolásán alapszik. Mivel mindegyik alrendszer saját →[adat]állományokkal és adatkezeléssel rendelkezik, a kapcsolatok megteremtéséhez →rendezések, →összeválogatások, átmeneti munka-[adat]állományok szükségesek.

Az →adatbázisokon alapuló ~nál az alrendszerek gyakorlatilag egymástól függetlenül működnek, az egységes rendszer alapja a közösen használt (fizikai és logikai) adatbázis.

integrált áramkör (integrated circuit) – olyan elektronikai komponens, amely nagyszámú diódát, tranzisztort, kondenzátort, ellenállást stb. tartalmaz ugyanazon a félvezető lapkán, a megfelelő kapcsolásokkal együtt, hogy jól meghatározott funkcionális egységet képezzen. Az alkatrészsűrűség alapján az ~öknek több generációját különböztetik meg: kisfokú integráltságú (10 v. kevesebb logikai kapu), középfokú integráltságú (10-100 logikai kapu), nagyfokú integráltságú (100-1000 logikai kapu), igen nagyfokú integráltságú (1000-10 000 logikai kapu), magasfokú integráltságú (10 000-100 000 logikai kapu) s legújabban az igen magas fokú integráltságú (0.1-5 millió logikai kapu) alkatrészsűrűséggel.

integrált információs rendszer (integrated information system) – olyan információs rendszer, amelynek szervezése a felhasználói alrendszerek egymástól függő, szoros kapcsolatán alapszik, megvalósítása az adatszerkezetek és az adatfeldolgozási műveletek integrációjára épül. Az adatok integrálása közösen használható →adatbázis előállításával valósul meg, az adatfeldolgozási műveletek integrálása pedig azok funkcionális szempontok szerinti egységes rendszerbe történő olyan összevonása, amely az ismételt műveleteket a minimálisra szorítja.

integrált szolgáltatású digitális hálózat (integrated services digital network) – teljes mértékben digitális, koaxiális kábelekre épülő nemzetközi kommunikációs hálózat, amely kép, hang, →fakszimile és számítógépes adatok átvitelére egyaránt alkalmas. A forgalmat két párhuzamos csatornán bonyolítja le: 64 Kbps sebességű adatcsatornán és 16 Kbps sebességű vezérlőcsatornán.

intelligens terminál (intelligent terminal) – korszerű, bizonyos feldolgozásokat is biztosító →terminál, amely saját vezérlőegységgel és →memóriával rendelkezik. Alapvető jelentősége, hogy akkor is képes feldolgozási műveleteket végezni, amikor nincs közvetlen kapcsolatban a központi számítógéppel. A →mikroprocesszorok elterjedésével az ~ok egy kisszámítógép minden lehetőségével bírhatnak, s ilyenkor a központi számítógéppel együtt ún. elosztott intelligenciájú rendszert alkotnak (→munkaállomás).

interaktív feldolgozás (interactive processing) – →információk rendszerint →párbeszédes üzemmódban történő olyan feldolgozása, amikor a felhasználó egy →terminálon keresztül állandó kapcsolatban van a számítógéprendszerrel, és a feldolgozást bármelyik pillanatban befolyásolhatja.

interaktív grafika (interactive graphics) – a →számítógépes grafika modern irányzata, amelynek alapvető eszköze a →grafikus kijelző. A rendszer →párbeszédes üzemmódban dolgozik, ami azt jelenti, hogy a gép éppen olyan gyorsan tud válaszolni a felhasználó igényeire, mint ahogyan a felhasználó reagál a gép válaszaira. A párbeszéd egyik oldalán a program áll, amely az eredményeket a képernyőn megjeleníti, a másik oldalon pedig a

felhasználó található, aki a képi információ alapján dönt a program további menetéről. A grafikus funkciók megvalósítására az →operációs rendszert megfelelő grafikus →szubrutinok csomagjával kell kiegészíteni, amelyeket a felhasználó →külső hivatkozások, parancsok formájában épít be alkalmazási programjába.

interaktív terminál (interactive terminal) – →párbeszédes terminál

interfész (interface) – →illesztőegység

intervallum [adat]típus (subrange [data] type) – →erősen típusos nyelvekben használatos, a programozó által definiálható elemi →sorszámozott [adat]típus. Értékkészletét valamely, már definiált elemi [adat]típus értékkészletének részhalmaza képezi úgy, hogy a programozó szabja meg az intervallum legkisebb, ill. legnagyobb értékét. Az – örökli az eredeti [adat]típus összes műveleteit.

invertált [adat]állomány (inverted file) – →adatbázisokkal kapcsolatban használatos olyan →[adat]állomány, amely az egy adott jellemzőhöz tartozó →entitásokat tartalmazza. A hagyományos [adat]állomány az entitásokat rögzíti (azok összes tulajdonságaival együtt), s így ez az adattárolási módszer az „egy adott entitásnak mik a jellemzői?" típusú lekérdezést teszi lehetővé. Ezzel szemben az – a „milyen entitások rendelkeznek az adott tulajdonsággal?" típusú válaszadásban használható. A teljesen – az összes jellemző minden egyes értékéhez tartozó entitásokat tartalmazza. A részlegesen – csak bizonyos jellemzők értékeivel kapcsolatos entitásokat tárol, mégis sokkal elterjedtebb a gyakorlatban.

irányított gráf (oriented graph) – →gráf

irodagépesítés (office automation) – →mikroszámítógépek, ill. →személyi számítógépek modern alkalmazási területe. Az irodai munkára jellemző szöveg-, kép-, ill. hangfeldolgozás és -átvitel feladatainak, valamint a hagyományos adatfeldolgozásnak integrált számítógépes megvalósításával foglalkozó alkalmazások összefoglaló megnevezése.

ismeret (knowledge) – olyan latens objektív →információ, amely tényekre, törvényekre, elvekre, fogalmakra, szabályokra, osztályozásokra, tapasztalatra, módszerekre stb. vonatkozik. Két szempontból is megközelíthető. Mint aktus azt az információáramlást jelenti, amely a megfigyelt egyedtől a megfigyelő felé halad. Mint latens információ azt a leíró v. definíciós információt jelenti, amely a megfigyelő egyed memóriájában rögzítődik. Ez a rögzítés magán viseli a megfigyelő egyed jellegzetességeit is, aki megszűri, rendszerezi a beérkező információkat, jelentéstartalmat kapcsol hozzájuk, kiértékeli a már régebben rögzített információkkal való esetleges kapcsolatokat stb.

ismeretábrázolás (knowledge representation) – bizonyos tárgykör →ismereteinek olyan rögzítési formája, amely a leginkább megkönnyíti a tárgykörben felmerülő feladatok megoldását. A →mesterséges intelligencia alapvető kutatási területe. →Szakértőrendszerekben gyakorlati megvalósításokat is felmutató legfontosabb formái: →szabály alapú, →logika alapú, →keret alapú, ill. →objektum alapú módszer.

ismeretbázis (knowledge base) – →ismereten alapuló programozási rendszerek (pl. →szakértőrendszerek) explicit formában elkülönített része, amely a →tárgyköri szakértő ismereteit tartalmazza. Lényegében speciálisan szervezett →memória, amely a tárgyköri tudást egy bizonyos módszer alapján rendezi (→ismeretábrázolás).

ismerettechnológia (knowledge engineering) – →ismeret alapú programrendszerek tervezésével, előállításával, felhasználásával kapcsolatos módszerek, eszközök, tevékenységek és tapasztalatok összessége.

ismerettechnológus (knowledge engineer) – →ismeret alapú programrendszerek tervezésében és megvalósításában részt vevő magas képesítésű számítástechnikai szakember, aki a →mesterséges intelligencia módszereinek felhasználásában is járatos. A →tárgyköri szakértő ismereteit, módszereit megfogalmazza, rendszerezi, strukturálja, megválasztja a specifikus ismeretek ábrázolásának legmegfelelőbb formáját, megtervezi és előállítja a legalkalmasabb következtetési mechanizmusokat stb.

ítélet (proposition) – a →matematikai logika egyik alapfogalma. Adott jelenségre, dologra vonatkozó olyan kijelentés, amely egyszerre v. csak igaz, v. csak hamis lehet.

iterációs eljárás (iterative method) – valamilyen feladat olyan típusú megoldása, ami-

kor ugyanazt a műveletsorozatot egymás után többször kell végrehajtani. Az egymást követő iterációk (ismétlések) során az előző menet eredménye lesz a következő iteráció kiindulási adata. Az első végrehajtás olyan kezdőértékkel történik, amely a megoldásnak közelítő értéke, a közbenső eredmények pedig a megoldás egyre jobb megközelítései (természetesen csak akkor, ha az eljárás konvergens).

A megoldást szolgáltató iterációk számát erősen befolyásolhatja a kezdőérték megfelelő megválasztása és az előírt pontosság.

A tárolt programú számítógépek egyik legelőnyösebb tulajdonsága, hogy a műveletsorozat ismételhető és az előre definiált v. dinamikusan változó feltétel elérésekor megállítható (→ciklus).

J

Jackson-módszer (Jackson's method) – a →lépésenkénti finomítás elvén alapuló programfejlesztési módszer. Központi elgondolása az, hogy a feldolgozandó adatok egymásutániságát, ill. az →adatstruktúrák természetét a feldolgozó programnak hűen vissza kell tükröznie a kiválasztott →vezérlőszerkezeteken keresztül. Segítségével jól strukturált, könnyen karbantartható, hatékonyan továbbfejleszthető programok állíthatók elő.

játék (game) – 1. Konfliktushelyzetek tanulmányozására alkotott matematikai modell. Formális, matematikai eszközökkel írja le az érdekellentétben lévő személyek (játékosok) megengedett lépéseit, a megoldást jelentő befejezés körülményeit és szabályait, a lehetséges kimenetelekhez vezető stratégiákat és a hozzájuk tartozó nyereségeket, ill. veszteségeket, az egyes lépések (akciók) által szerezhető információ jellegét és mennyiségét. Ha a ~ kimenetele emberi döntések által befolyásolható, stratégiai ~ról, máskülönben szerencse~ról beszélnek. Az antagonisztikus ~ban csak két, egészében ellentétes érdekeltségű játékos vesz részt (tehát az egyik nyeresége pontosan akkora veszteséget jelent a másiknak). A véges ~ban véges számú játékos vehet részt, minden lépésben véges számú lehetőség közül lehet választani, és a ~ véges számú lépés után biztosan befejeződik. – 2. Számítógépes játékok, szórakoztató programok.

játékelmélet (games theory) – érdekellentétes helyzetek tanulmányozásával, az optimális megoldáshoz vezető stratégia szabályaival foglalkozó matematikai elmélet. Magát a konfliktust →játéknak, az érdekellentétben álló résztvevőket játékosoknak nevezik. A ~ megalapozója Neumann János, akinek 1928-ban jelent meg első munkája a témával kapcsolatban, majd 1944-ben, O. Morgensternnel közösen kiadott tanulmánya (Játékelmélet és gazdasági viselkedés) jelentette az új matematikai diszciplína megjelenését.

játékfeladat (toy problem) – a →mesterséges intelligencia elveinek, eljárásainak, programjainak kipróbálására, bizonyítására szolgáló egyszerű feladat, amely önmagában gyakorlati jelentőséggel nem rendelkezik. Rendszerint valamilyen szellemi játéknak v. a valós világban létező összetett feladatnak megközelítő, leegyszerűsített változata.

jel (signal) – valamely fizikai jelenség állapotváltozásait jellemző, azonosítható és közlésre alkalmas információ, amelyhez bizonyos jelentés és/vagy hatás fűződik. A modern számítástechnikában szinte kizárólag elektromos, optikai és mágneses ~eket használnak. Az analóg technikában rendszerint folytonos áramerősség v. feszültség, a digitális technikában pedig két jól megkülönböztethető feszültségszint használatos, amelyekhez logikai értékek kapcsolódnak.

jelentésmegmaradás elve (semantic invariance principle) – →fordítóprogramok, ill. →értelmezőprogramok megvalósíthatóságának alapelve, amely szerint egy program szemantikai jelentése nem változhat meg a rajta végzett átalakítások során. A program jelentésén a végrehajtás által a feldolgozandó információra és a számítógépre magára gyakorolt hatást értik. A jelentés tehát a számítógép segítségével, a gép működése során jut érvényre. A programok jelentését még a végrehajtás előtt meg kell határozni, le kell írni. E műveletek elvégzésére szolgálnak a →programozási nyelvek, viszont az ezekkel kódolt programokat a számítógép által elvégezhető formára kell alakítani, s ehhez többszöri átalakítás szükséges.

jelfogós számítógép (relay computer) – a tulajdonképpeni elektronikus számítógépek előhírnökének tekintett, a harmincas évek végén gyártott számítógéptípus. A kétállapotú elemi tárolóknak a jelfogók meghúzott v. elengedett állapota felelt meg. Az első programvezérlésű számítógép ~ volt, K. Zuse készítette 1941-ben.

jelsorozat (string) – →karakterlánc

jelszó (password) – olyan egyedi azonosító, amelynek ismeretében bizonyos →erőforrás, adathalmaz elérhető, vagyis az az eszköz, amelynek révén a számítógéprendszerekben tárolt információk védelme biztosítható.

jelző (flag) – programozástechnikai értelmezésben olyan, az adatfeldolgozásban közvetlenül részt vevő adathoz rendelt járulékos információ, amely a feldolgozásban közvetlenül nem vesz részt, hanem annak termé-

szetére, esetleg különböző formáinak kiválasztására utal. Pl. →logikai rekordhoz rendelt ~ lehet az az információ, hogy a rekordhoz tartozó adatok előzetes ellenőrzése során fellépett-e valamilyen hiba. A rekordnak az →[adat]állományba való beillesztését ennek a ~nek az alapján lehet elvégezni v. letiltani. Ugyancsak ~k segítségével lehet rögzíteni azoknak a pillanatnyi körülményeknek az állapotát, amelyek a későbbiekben, a program végrehajtása során meghatározzák egyik v. másik programág kiválasztását. →Kötegelt feldolgozásnál speciális ~k megfelelő beállításával lehet a munka következő lépéseit szelektíven végrehajtani, esetleg felfüggeszteni.

jobbra igazodó ábrázolás (right justified representation) – →alfanumerikus adatok (→jelsorozatok) olyan tárolási formája, amelyben az adat felépítő karakterei a számukra fenntartott →bájtokat a zóna jobb első bájtjától kezdve töltik ki. Ha a karakterek száma kisebb, mint a kiosztott zóna bájtjaié, az üres helyeket szóközök töltik ki a zóna bal széléig (szemben a →balra igazodó ábrázolással). Magas szintű programozási nyelvekben speciális előírást használnak a ~ jelzésére (ilyen pl. a →COBOL nyelvben a JUSTIFIED RIGHT).

K

K – számítástechnikai vonatkozásban mértékegységek (tárolási kapacitás, átviteli sebesség stb.) kifejezésére használt egyezményes jel, a kilo rövidítése, értéke $2^{10}=1024$ (pl. Kbájt=1024 bájt).

kapcsolás (linking) – az a művelet, amelynek fő feladata a különböző →tárgymodulok közötti →külső hivatkozások (szimbolikus hivatkozások) feloldása, a →modulokhoz tartozó →névtartományok egyetlen →címtartományba való átalakítása, valamint az esetleges →átfedési szerkezet megvalósítása. Az →operációs rendszerek speciális rendszerprogramokat tartalmaznak a ~ végrehajtására (→kapcsolóprogram).

kapcsolóprogram (linkage editor) – →operációs rendszerekhez tartozó rendszerprogram, amely az egy v. több →fordítás útján létrejött →tárgymodulokat egyetlen →tárgyprogrammá építi fel, végrehajtja a →kapcsolás műveletét.

kapcsolt hálózat (switched network) – összefoglaló megnevezése azoknak az →adatátviteli hálózatoknak, amelyeket felhasználói megosztva használhatnak úgy, hogy az összeköttetés két-két →adatállomás között bizonyos hálózatelemek időleges egymáshoz kapcsolása révén épül fel (→vonalkapcsolás, →üzenetkapcsolás, →csomagkapcsolás).

kapcsolt virtuális összeköttetés (switched virtual circuit) – →csomagkapcsolásos →adatátviteli hálózatokban két →adatvégállomást úgy összekötő →kommunikációs csatorna, hogy az összeköttetés csak az átvitel időtartamára érvényes (→kapcsolt vonal).

kapcsolt vonal (switched line) – az →adatátviteli hálózatokban részt vevő →adatállomások között (légvezetékek, kábelek, rádió-összeköttetések, kapcsolóközpontok és egyéb átviteli berendezések révén) létrehozott ideiglenes fizikai kapcsolat, amely mint olyan, csak az átvitel idején létezik (→üzenetkapcsolás, →csomagkapcsolás).

karakter (character) – a felhasznált jelkészlet egy elemének (betű, szám, speciális szimbólum) megfelelő bitvektor, amely önálló egységként vesz részt az adatátvitelben v. -feldolgozásban. A →bájtszervezésű számítógépek →operatív memóriájában egy ~ 1 bájton ábrázolható. Az angol ábécé betűit alfabetikus ~eknek nevezik; ha ezekhez hozzáteszik a tíz decimális számjegyet és bizonyos számú speciális jelet, együttesen alfanumerikus ~készletet alkotnak.

karakter [adat]típus (character [data] type) – →erősen típusos nyelvekben használatos →előre definiált, →sorszámozott [adat]típus →karakterek kezelésére. Értékkészlete rendszerint az →ASCII-kód 128 (v. 256) karaktere, a megengedett műveletek az értékadás és az összehasonlítások.

karakterfelismerés (character recognition) – géppel v. kézzel írott szövegek letapogatására és felismerésére szolgáló eljárás. Kezdetben a szövegeket szabványos betűtípusokkal és mágnestintával kellett írni, ma már léteznek módszerek a közönséges nyomtatással, sőt kézírással rögzített szövegek felismerésére is.

karakterisztika (characteristic) – →lebegőpontos számábrázolással kapcsolatos fogalom. Ennél az ábrázolási módnál két előjel tárolására volna szükség: egy a →mantissza, egy pedig a kitevő számára. Ezt a hátrányt úgy küszöbölik ki, hogy csak a mantissza előjelének tartanak fenn bitpozíciót (rendszerint az ábrázolási bitvektor bal szélső bitjét), az exponens helyett pedig a ~t ábrázolják. Ha az exponens számára fenntartott bitpozíciók száma 7, akkor ennek modulusa 128. A ~t tehát úgy határozzák meg, hogy a legnagyobb pozitív érték 128 legyen:

$$C = 64 + E$$

ahol E az exponens értéke. Ha a mantissza negatív, a ~ az abszolút értékének komplementuma lesz. Innen következik, hogy lebegőpontos ábrázolásnál a negatív szám az egész ábrázolási vektor komplementuma. Ezzel a módszerrel elérhető az, hogy az összehasonlítás és a komplementumképzés műveleteit a →fixpontos műveletekkel lehessen elvégezni.

karakterisztika-alulcsordulás (characteristic underflow) – →lebegőpontos számábrázolásnál előállható különleges helyzet, amikor az ábrázolandó mennyiség olyan kicsi (tehát a kitevő olyan nagy abszolút értékben), hogy „nem fér bele" a →karakterisztika szá-

mára fenntartott bitpozíciókba. A ~ →csapdát eredményez, és a program végrehajtása megszakad.

karakterisztika-túlcsordulás (characteristic overflow) – →lebegőpontos számábrázolásnál előállható különleges helyzet, amikor az ábrázolandó mennyiség olyan nagy (tehát a kitevő olyan nagy abszolút értékben), hogy „nem fér bele" a →karakterisztika számára fenntartott bitpozíciókba. A ~ →csapdát eredményez, és a program végrehajtása megszakad.

karakterkészlet (character set) – 1. Bizonyos formában kódolt olyan karakterek halmaza, amelyeket adott számítógéprendszer felismer. Pl. igen elterjedt a 64 karakteres ~, amelyhez a 10 decimális számjegy, az angol ábécé 26 betűje, valamint 28 speciális karakter tartozik. Szintén gyakran használt az →ASCII-kódolású, ill. az →EBCDIC-kódolású, 128 v. 256 szimbólummal rendelkező ~. – 2. →Nyomtatón v. →megjelenítőn előállítható karakterek halmaza.

karakterlánc (character string) – jól meghatározott karakterhalmaz elemeinek változó v. fix hosszúságú sorozata. Az →operatív memóriában folytonosan tárolható, vagyis nincs szükség kiegészítő információkra a sorolást illetően. Egyes magas szintű programozási nyelvek speciális utasításokat tartalmaznak a ~ok kezelésére. (Szinonimái: jelsorozat, karakterfüzér.)

karbantartó program (maintenance program) – a →kezelőprogramok kategóriájába tartozó olyan speciális program, amellyel a számítógéprendszer különböző hardver- v. szoftveregységeinek helyes működését v. épségét lehet ellenőrizni, ill. a szoftveregységek hibáit lehet korrigálni, esetleg bizonyos változtatásokat lehet végrehajtani az →operációs rendszer felépítésében.

kártyalyukasztó (card puncher/perforator) – 1. Számítógéprendszer olyan →kimeneti egysége, amely az →operatív memóriában tárolt, egyezményesen kódolt információt átalakítja és azt a →lyukkártyákra lyukasztja. – 2. Lyukkártyák lyukasztására szolgáló, klaviatúrával ellátott berendezés, amelynek segítségével a felhasználók – a számítógép részére bemenő – információkat kódolhatják. A lyukasztások könnyebb ellenőrzésére a legtöbb ~ értelmezőegységgel is el van látva, amely az egyes oszlopok perforációinak megfelelő karaktert a kártyára ki is nyomtatja.

kártyaolvasó (card reader) – számítógéprendszer olyan →bemeneti egysége, amely a →lyukkártyákon perforációkombinációk formájában rögzített információt fotoelektromos úton érzékeli és a számítógép által közvetlenül felhasználható, egyezményes formára alakítja.

kártya-szalag konverter (card-to-tape converter) – elektronikus berendezés v. számítógépes program, amely a →lyukkártyákon perforációkombinációk formájában rögzített információt →lyukszalagra v. →mágnesszalagra jellemző kódra alakítja át és rögzíti.

katalogizált eljárás (cataloged procedure) – →kötegelt feldolgozásnál használt eljárás. Lényegében egy →munka különböző lépéseinek (fázisainak) automatikus láncolását teszi lehetővé azáltal, hogy a lépésekre jellemző →vezérkártyákat speciális →forrásnyelvű könyvtárban rögzítik. A standard →bemeneti egységen megjelenő egyetlen vezérkártya, amely a ~ra utal, ezt az egységet automatikusan átkapcsolja a jelzett könyvtárra, s onnan folytatódik a további munkalépések láncolása. Egyes →operációs rendszerek lehetővé teszik paraméterezett ~ok felhasználását is.

katalógus (catalogue) – az →operációs rendszerek értelmezésében olyan táblázat, amely a →könyvtár jellemzőit és a könyvtárban található programokkal kapcsolatos információkat tartalmazza.

katódsugárcsöves megjelenítő (cathod ray tube displaying device) – számítógéprendszerekben – különösen →párbeszédes üzemmódban – alfanumerikus karakterek és/vagy grafikus ábrák kijelzésére alkalmas korszerű →megjelenítő. Rendszerint billentyűzettel van ellátva, ezáltal adatbevitelre is használható. →Párbeszédes terminálként a felhasználó és a számítógéprendszer közötti információcsere alapvető eszköze, de sok →kötegelt feldolgozású számítógépnél használják a →konzolíró gép helyettesítésére is. A grafikus ~ →fényceruzával kiegészítve szintén alkalmas adatbevitelre.

kazettás mágneslemez (cartridge disk) – kisszámítógépes v. mikroszámítógépes rend-

szerek gyakori →háttértára. Kazettában elhelyezett, azzal együtt cserélhető, 2.5-25 Mbájt kapacitású mágneslemez.

képernyő (screen) – →megjelenítő

képernyőadapter (screen adapter) – →képernyőn megjelenített képek létrehozását támogató →bővítőkártya.

képfeldolgozás (image processing) – a →számítógépes grafika egyik fontos felhasználási területe. Fontosabb műveletei: a képek megjelenítése, tárolás céljából történő kódolása, visszaállítása; osztályozása, elemzése stb. A képek belső ábrázolása rendszerint egy olyan mátrix, amely az elemeire bontott kép minden egyes pontjára vonatkozó információkat rögzíti. A ~ legfontosabb alkalmazásai: a műholdakról és repülőgépekről készített felvételek kiértékelése; a felhőzet, az ásványi kincsek eloszlása, a növénytakaró kiértékelése; térképek készítése; képi információk koncentrált tárolása; nehéz v. különleges körülmények között készült képek és tomogramok kiértékelése stb.

képfrissítés (image refreshing) – képek jó minőségű megjelenítésére használt módszer. Lényege, hogy a megjelenítő áramkör ciklikusan vetíti a képernyőtár tartalmát (→bittérképezés) a képernyőre. Hogy a szemet fárasztó →villogást kiküszöböljék, a ~ frekvenciaértékének legalább 50-60 Hz-nek kell lennie.

képlethiba (error of formula) – matematikai modellel definiált és numerikus módszerrel (a módszer által használt képlettel) elvileg számítható értékek közötti különbség, ugyanis a numerikus eljárások általában a matematikailag megfogalmazott modellnek csak egy közelítését adják meg. Ezért a numerikus módszereknek →hibaképletek v. →hibakorlátok formájában tartalmazniuk kell a ~ra vonatkozó becsléseket.

képpont (pixel) – →képernyőn v. →nyomtatón előállított kép elemi része (a képernyőn fénypont, nyomtatónál általában festékes nyomtatott pont), amely közvetlen vezérlés alatt áll. A hordozó felület mm^2-ein megvalósítható ~ok száma az eszköz egyik igen fontos jellemzőjének, a →felbontásnak mértéke.

kérdezés karakter (enquiry character) – →szinkron adatátviteli eljárásokban használt →vezérlőjel. Amikor a vezérállomás a vonalra adja a lekérdezendő →terminál vezérlőegységének és a terminálnak a fizikai címét, adását ~rel zárja. A ~ válaszra szólítja fel a lekérdezett terminált, hatására az adatátviteli vezérállomás vétel üzemmódra vált át.

kerekítési hiba (round-off/rounding error) – az a különbség, amely a gépi és a pontos számítás eredményei között kialakul. Ez abból adódik, hogy a számítógép az előírt műveleteket csak véges számú jeggyel ábrázolt adatokkal hajtja végre. Ezért a numerikus algoritmusok számítógépre vitelénél nagy figyelmet kell szentelni a →hibabecslésnek, és a megfelelő javítási képletek segítségével ki kell küszöbölni a ~kat, ellenkező esetben azok erősen felhalmozódhatnak és meghamisíthatják az eredményt.

keresés (searching) – olyan algoritmus, amelynek célja adott tulajdonsággal (→kulcsértékkel) rendelkező elem kiválasztása az adatstruktúrából (→[adat]állományból, →táblázatból), v. annak egyértelmű megállapítása, hogy a kérdéses adatelem nincs az illető adatstruktúrában. A ~ szempontjából az adatelemnek csak a kulcsa lényeges, a többi felépítő mező tartalma közömbös. Táblázatokban végzett ~ alapvető módszere a →soros (lineáris) és a →bináris (logaritmikus) keresés.

keresési kulcs (search key) – olyan →kulcs, amelynek alapján egy →heterogén adatstruktúra (→[adat]állomány v. →táblázat) jól meghatározott adatelemeit ki lehet választani.

keresés táblázatban (table search) – a →keresés műveletét magas szintű programozási nyelvekben megvalósító utasítás (pl. SEARCH a →COBOL nyelvben). →Táblázat (esetleg →tömb) szerkezetű struktúrákban adott tulajdonsággal rendelkező adatelem lokalizálására használható. Sikeres végrehajtása után a kikeresett adatelem címe a további feldolgozás rendelkezésére áll az adatstruktúrában.

keresztasszembler (cross-assembler) – olyan →asszembler, amely egy fejlesztés alatt álló számítógép asszembler nyelvén megírt programot lefordít annak →gépi nyelvére, és emellett képes egyéb információkat is szolgáltatni a programozó számára. Járulékos funkciói közé tartozik a →szintaktikai hibák felfedezése és kijelzése, a →szimbólumtáblázat

elkészítése, a programmal kapcsolatos statisztikai információk közlése stb. A – a →keresztszoftver egyik legfontosabb összetevőjeként annak általános jellemzőivel is rendelkezik, tehát elsősorban fejlesztő számítógépen működik, →tárgyprogramját csak a célszámítógép végezheti el (a fejlesztő gépen a végrehajtást esetleg szimulálni lehet).

keresztfordító (cross-compiler) – speciális →fordítóprogram, amely egy számítógépen működik, de →tárgyprogramját csak egy másik számítógép végezheti el közvetlenül. Felhasználása két esetben jelent előnyt: ha a célszámítógépen nincs meg az illető programozási nyelv fordítóprogramja, v. ha a célszámítógép még csak a fejlesztés periódusában van. Ez utóbbi esetben elérhető, hogy a →hardver és a →szoftver párhuzamosan készüljön az új számítógéprendszer számára. A – különleges jelentőséget nyert a →mikroprocesszoros programfejlesztésekben, ugyanis gyakorlatilag bármilyen mikroprocesszor utasításainak a fogadására alkalmassá tehető az utasításkészlet-táblázat kicserélésével, ill. kiegészítésével (esetleg további módosításokra is szükség lehet azért, hogy a tárgyprogramkimeneten a lefordított program a mikroprocesszor által megkívánt formában álljon elő).

kereszthivatkozások listája (cross reference list) – →fordítóprogramok (ill. →assemblerek) opcionális szolgáltatásaként előállított olyan lista, amely a lefordított programban használt →azonosítók, szimbólumok definiálási és felhasználási helyeit rendszerezett formában rögzíti. A – igen hasznos segédeszköz a programok belövési és tesztelési fázisában.

keresztszoftver (cross-software) – speciális rendszerprogramok gyűjteménye, amelyek már létező számítógépen működnek azzal a feladattal, hogy egy, még fejlesztés alatt álló számítógépre →szoftvert dolgozzanak ki. Legfontosabb komponensei a →keresztassembler, a →keresztfordító, a célszámítógép működését szimuláló programok, szövegszerkesztők stb. Nagy előnye, hogy a különböző programfunkciókat még a fejlesztés alatt álló berendezés megszületése előtt ki lehet próbálni, a programok logikailag ellenőrizhetők és javíthatók, és mindezekhez hatékony segédeszközökkel rendelkezik (→ellenőrzött végrehajtás, →lépésenkénti végrehajtás).

keret (frame) – 1. A →mágnesszalag olyan keresztirányú zónája, amely egy karakter (7-9 bit) rögzítésére alkalmas. A rögzítési sűrűség gyakori egysége a hüvelykenkénti –szám, gyakorlati értékei 200, 556, 800, 1600, 6000 keret/hüvelyk. – 2. →Virtuális memóriával kapcsolatos fogalom: az →operatív memóriának az az egysége, amely a →lapkezelési technikában egy logikai →lapnak felel meg (szokták még blokknak v. fizikai lapnak is nevezni). – 3. →Időosztásos multiplexelést használó hálózatokban az események teljes ciklusa.

keret alapú módszer (frame-based method) – →ismeretábrázolásban használható olyan programozási módszer, amelyben a tényeket, fogalmakat csomópontok képviselik, s ezekhez tulajdonságokat lehet kapcsolni. A csomópontokat relációk (összefüggések) kapcsolják össze, s így hierarchikusan szervezett hálót alkotnak, amelyben a tulajdonságok átöröklődhetnek.

késleltetési vektor (delay vector) – →csomagkapcsolásos →adatátviteli hálózatokban az →alkalmazkodó adatútképzés egyik megvalósítási eszköze. Adott csomópontra vonatkoztatva az összes többi számára szánt adatcsomagok becsült átviteli időit tartalmazza. Az adaptív adatútképzés az optimális adatutat a minimális késleltetés alapján választja ki.

kétcímű számítógép (two-address computer) – olyan →digitális számítógép, amelynek utasításaiban két →cím szerepel. Ezek rendszerint a művelet két operandusának címeit jelölik, a művelet eredményét általában az egyik operandus címére helyezik (természetesen annak tartalmát felülírva).

kétértelmű szabály (ambigous rule) – →döntési táblázatokban előfordulható olyan eset, amikor – a feltételek nem független természete miatt – a feltételértékek egymásnak ellentmondanak.

kétirányú lista (two-way list) – olyan →lista, amelyben minden adatelemnek egyetlen követője és egyetlen megelőzője van, vagyis az adatstruktúra bármelyik eleme elérhető az első v. az utolsó adatelemből kiindulva.

kétszeresen láncolt lista (doubly linked list) – →kétirányú lista

kezdőértékadó utasítás (data initialization statement) – a legtöbb →programozási nyelvben előforduló olyan →nemvégrehajtható utasítás, amelynek segítségével a programozó egyes programváltozók számára kezdőértéket határozhat meg, még a program végrehajtása előtt. (Pl. DATA a →FORTRAN-ban, INITIAL a →PL/1-ben, VALUE a →COBOL-ban stb.). Mint nemvégrehajtható utasítás, a generált kódban nem hagy közvetlen nyomot, viszont a →fordítóprogram az illető változók számára fenntartott memóriaterületen az igényelt értékeket generálja. Ez azt jelenti, hogy a →tárgyprogramnak az →operatív memóriába való betöltése után az érintett változók az előírt értékekkel fognak rendelkezni.

kezdőszimbólum (goal/sentence symbol) – →generatív nyelvtan formális definíciójában szereplő, kiemelt jelentőségű →nemterminális szimbólum, amelyből →helyettesítési szabályok ismételt és megfelelő alkalmazásával az adott nyelvhez tartozó bármelyik →mondatforma, ill. →mondat előállítható.

kezelőprogram (utility program) – konkrét számítógéprendszer →programkönyvtárában megtalálható olyan program, amely a felhasználók rendelkezésére áll a programfejlesztés egyes rutinfeladatainak végrehajtásában. A legáltalánosabban ismert ~ok a másoló, a szövegszerkesztő, a leltározó, a könyvtárkezelő, a rendező stb. programok. (Szinonimája a segédprogram.)

kézfogás (handshaking) – →kölcsönös szinkronizálás

kiadványszerkesztő (desktop publishing) – →munkaállomásokra kifejlesztett speciális programtermék kiadványok szerkesztésére, amely a specifikus nyomdai műveleteket támogatja (beleértve nemcsak szövegek, hanem grafikus ábrák, képek stb. kezelését is).

kibernetika (cybernetics) – bonyolult rendszerek törvényszerűségeit (működését, szabályozási és vezérlési szabályait, információértelmezési és -feldolgozási lehetőségeit stb.) természettudományos, ill. matematikai módszerekkel vizsgáló tudomány. Megalapozója Norbert Wiener.

Egy ~i rendszer alapvető eleme a bemenete és kimenete közötti információs kapcsolat, az ún. visszacsatolás. Tekintve, hogy a vezérlési folyamatokat információáramlási szempontból vizsgálja, a rendszerek fizikai természetétől függetlenül, lehetségessé teszi a különböző természetű – biológiai, fizikai, technikai,társadalmi stb. – rendszerek egységes tanulmányozását.

kicsomagolás (unpacking) – 1. →Csomagkapcsolásos →adatátviteli hálózatokban végzett művelet, amelynek révén a vezérlési adatok elhagyásával az értékes információ visszanyeri eredeti formáját. – 2. →Tömörített decimális ábrázolású adatok átalakítása →zónázott decimális ábrázolásra. – 3. Tömörítőprogramok által bepakolt adatok eredeti formájának helyreállítása.

kiegészítő rekord (additional record) – olyan →logikai rekord, amely az →[adat]állomány feldolgozásánál kerül az adathalmazba úgy, hogy nem helyettesített benne egy már jelen lévőt (→indexszekvenciális [adat]állomány, →direkt szervezésű [adat]állomány). A ~ok számára rendszerint túlcsordulási területet kell fenntartani.

kiegyensúlyozott fa (balanced tree) – olyan →fa, amelynél minden egyes csomóponthoz ugyanannyi számú ág tartozik, és a fa elágazási kapacitása fentről lefelé haladva, minden sorban balról jobbra lépve ki van használva. A ~ speciális fajtája a bináris ~, ahol csomópontonként maximálisan két ág lehetséges. Jelentősége elsősorban az aritmetikai kifejezések →operatív memóriabeli ábrázolásában van.

kieséses rendezés (tournament sorting) – a →rendezésnek az →operatív memóriában megvalósítható módszere. Lényege: a rendezendő halmaz elemeit összepárosítják, és a „győztes" (a kisebb v. a nagyobb) jut tovább a következő fordulóba. Itt újra összepárosítják őket, és ismét a győztesek jutnak tovább, egészen a döntőig, amelynek győztese a legjobb (legkisebb v. legnagyobb) elem. (Az eljárás és elnevezése a kieséses rendszerű kupamérkőzésektől származik.)

kihasználási együttható (service factor) – →erőforrás foglaltsági szintjét kifejező jellemző. Értékét az időegység alatt kiszolgált és a jelentkezett összes igény hányadosa adja.

kiírási hiba (writing error) – adatoknak az →operatív memóriából külső →adathordozóra történő kiírása közben fellépő hibatípus,

amely a →multiplexcsatorna, a →szelektorcsatorna, ill. a →periferikus mű hibás működésének következménye. Kiszűrése elsősorban az →operációs rendszer feladata, amely a →központi vezérlőegységen keresztül állandó felügyelet alatt kell hogy tartsa a kiviteli művelet végrehajtását. A ~ rendszerint a program végrehajtásának rendellenes befejezéséhez vezet. Mágneses adathordozónál, ahol a kiírási művelet megismételhető, az operációs rendszer csak többszöri próbálkozás sikertelensége után jelez ~t.

kijelentkezés (log off) – →párbeszédes üzemmódban dolgozó rendszerekben a felhasználó jelzése az →operációs rendszernek, hogy már nem kívánja igénybe venni annak szolgáltatásait. A ~ következtében a felhasználó esetleg még aktív programjai befejeződnek, a lefoglalt →erőforrások pedig felszabadulnak.

kijelzés (displaying) – számítógéprendszer által szolgáltatott, a felhasználók részére közvetlenül olvasható (v. értelmezhető) információ. Alapvető típusai az →alfanumerikus kijelzés és a →grafikus kijelzés.

kijelző (displaying device) – →megjelenítő

kimeneti [adat]állomány (output file) – az az →[adat]állomány, amely a feldolgozás útján nyert adatokat tartalmazza. Különleges jelentőségű →kötegelt feldolgozásnál a standard ~, amely a →háttérpartícióban végrehajtott programok kimenő adatait, valamint az →operációs rendszer egyes üzeneteit hordozza. Ugyancsak ebbe kerülnek a →fordítóprogramok, →kapcsolóprogramok és más →rendszerprogramok hibaüzenetei, a →memóriakiíratás eredménye stb. Modernebb rendszerek megkülönböztetnek standard ~t és hibaállományt, noha mindkettő rendszerint ugyanott jelenik meg a képernyőn v. a nyomtatón, de bármelyik leválasztható és máshova küldhető.

kimeneti egység (output unit) – olyan, a számítógéphez kapcsolt →periferikus mű, amely adatoknak az →operatív memóriából történő kihozatalára alkalmas. Lehet önálló egység (pl. →kártyalyukasztó, →szalaglyukasztó stb.), és lehet alrendszere is egy →b/k egységnek (pl. mágneslemez-meghajtó, mágnesszalag-meghajtó). Az →operációs rendszer szempontjából kiemelt szerepe van a standard ~nek, amely a →háttérpartíció dedikált erőforrása. Ezen keresztül az operációs rendszer a feldolgozás nyomán nyert adatokon kívül a rendszer működésével kapcsolatos üzeneteket, hibajelzéseket is közölhet a felhasználókkal. Rendszerint speciális [adat]állományként kezelik (→kimeneti [adat]állomány).

kimeneti függvény (output equation) – az a logikai függvény, amely meghatározza egy logikai hálózat kimenetein megjelenő jel értékét. Szekvenciális logikai hálózatnál a ~ben megjelennek a bemeneti logikai változók mellett a hálózat állapotait meghatározó változók is, míg a kombinációs hálózatoknál a ~ csak a bemeneti logikai változókat tartalmazza.

kimenő ábécé (output alphabet) – valamely →automata kimenetén válaszként nyerhető szimbólumok halmaza v. →fordítóprogramok által előállított →tárgyprogramok szimbólumainak összessége.

kiterjedt hálózat (wide area network) – nagyobb földrajzi területeket lefedő (általában 1 km sugarú körzeten kívüli) →adatátviteli hálózat. Alapvető jellegzetessége, hogy az átvitelt analóg hordozók segítségével (tehát →modemek közreműködésével →modulálással) valósítják meg.

kiterjesztett gép (extended machine) – az a számítógép, amely a felhasználó szempontjából az →operációs rendszeren keresztül látható. Más megközelítésben olyan gép, amely az →alapgép utasításain kívül az ún. →felügyelőprogram-hívó utasítások felismerése révén a felhasználó rendelkezésére bocsátja a →szoftver erőforrásokat is.

kiterjeszthető nyelv (extensible language) – új típusú →programozási nyelvek kifejlesztésének egyik fontos célkitűzése. A ~ lényege az, hogy csak egy kicsi, minden feladathoz feltétlenül szükséges magot (bázisnyelvet) definiál, de ugyanakkor megfelelő eszközöket biztosít nyelvi kategóriák definiálására (pl. →FORTH, LOGO stb.).

kitüntetett (privileged) – olyan felhasználó v. program jelzője, aki v. amely az →operációs rendszer hatáskörébe eső műveleteket végezheti, és az egész rendszer működését korlátozas nélkül befolyásolhatja.

kiválasztásos rendezés (selection sorting) – a →rendezésnek az →operatív memóriában elvégezhető módszere. Lényege: kivá-

lasztják a rendezendő halmaz legkisebb (v. legnagyobb) elemét, és átviszik a rendezett halmaz első helyére; a következő lépésben ugyanezt a műveletet hajtják végre, de most már a kiválasztott elem nélküli halmazon. Ezt egészen addig folytatják, amíg a rendezetlen halmaz minden elemét át nem vitték a rendezett halmazba.

kivétel (exception) – →valós időben, potenciális hibák között dolgozó programok végrehajtása közben előforduló helyzet, amely kompromittálhatja v. meggátolhatja a további rendes végrehajtást. Az ilyen programok rendellenes befejeződése viszont gyakorlati szempontból elfogadhatatlan (ilyen pl. egy távbeszélő kapcsolóközpontot vezérlő program, amely hibás szám beütésére leállna!), ezért a modern programozási nyelvek (pl. →CHILL, →Ada) megfelelő eszközökkel rendelkeznek az ilyen ~ek kezelésére. Elvi szempontból a ~ hasonlít a →csapdához, de amíg ez utóbbi a végrehajtó rendszer (számítógép) belső hibáira reagál, addig a ~ a vezérelt környezet esetleges rendellenességeire vonatkozik.

kivitel (output) – az →operatív memóriában tárolt információnak a számítógéprendszerhez tartozó →periferikus művön v. →kimeneti egységén keresztül külső →adathordozóra történő rögzítése.

kizáró feltételrendszer (exclusive condition system) – a →döntési táblázatokban előforduló olyan eset, amikor az adott feltételek közül csak egy lehet igaz. Pl.

A>B	Y	–
A<B	–	Y

kizáró VAGY (exclusive OR) – olyan kétargumentumú →logikai függvény, amely akkor és csakis akkor igaz, ha a két argumentum közül egy és csakis egy igaz. Igazságtáblázata:

kVagy	0	1
0	0	1
1	1	0

klauzula (clause) – valamely →programozási nyelvhez tartozó →kulcsszók rendezett sorozata, amely egy →mondat →szemantikai tartalmának valamely részét fejezi ki.

kocsi (carriage) – egyes írásra v. nyomtatásra szolgáló berendezéseknél az a mozgó szerkezet, amely rendszerint az írófejet hordozza. Mozgását az egyes karakterek leütése határozza meg, a nagyobb távolságokra lévő rögzített íráshelyekre a tabulátorberendezéssel mozgatható. Az utolsó nyomtatható karakter leütése nyomán a ~ rendszerint – soremelést is eredményezve – visszaugrik az első nyomtatható oszlopra. Ugyanezt a műveletet speciális billentyű leütése is kiválthatja (→„kocsi vissza" karakter). Ezek a műveletek logikai szinten a →katódsugárcsöves kijelzőknél is megmaradtak, ahol már nincs tulajdonképpeni ~.

„kocsi vissza" karakter (carriage-return character) – kijelzésre v. írásra szolgáló berendezések funkcionális billentyűje. Kezelése soremelést és az új sor bal első pozíciójára ugrást idéz elő. Egyben →vezérlőjel is a →terminálok vezérlésére, a bevitel lezárását is jelzi.

kód (code) – 1. Bizonyos megállapodásszerű, egyezményes jelek, ill. jelkombinációk rendszere, amelynek segítségével információ kifejezhető v. ábrázolható. Tágabb értelemben egy információ minden megjelenési formája valamilyen ~nak felel meg. – 2. Szövegösszefüggésekben: a fenti rendszer egyedi alkalmazásának az eredménye, vagyis maga a rendszerre jellemző jelekkel kódolt (ábrázolt) információ.

kódgenerálás (code generation) – programok →fordításának alapvető fázisa, amely a →szintaktikai analízis, a →szemantikai analízis és a →memóriakiosztás alapján előállítja a →tárgyprogramot. Végrehajtásának központjában a →jelentésmegmaradás elve áll: a →fordítóprogram a felismert forrásnyelvű elemekből kiindulva tárgynyelvű elemcsoportokat állít elő. Az előállított tárgyprogram lehetséges formái: közvetlenül végrehajtható →gépi nyelvű program; a →kapcsolóprogram által még feldolgozandó →áthelyezhető bináris formájú programegység(ek); →asszembler nyelvű program, amelyet még az →asszemblernek tovább kell alakítania; egy olyan nyelvű program, amely további fordítás útján válik csak végrehajthatóvá.

kódkonverter (code converter) – olyan berendezés (esetleg számítógépes program), amely az információ egyik kódolt formá-

jából ugyanannak az információnak egy másik kódolt formáját állítja elő.

kódolás (encoding) – 1. Valamely karakterkészlet (ábécé) minden egyes szimbólumához egy másik karakterkészlet jól meghatározott karakterének egyértelmű hozzárendelése. – 2. Programtermékek előállításának az a fázisa, amelyben a számítógép által elvégzendő feladatokat (forrásprogramokat) valamilyen →programozási nyelv eszközeivel leírják.

kódoptimalizálás (code optimization) – →tárgyprogramok kézi v. automatizált eszközökkel végzett optimalizálása. Az optimalizálás kritériuma v. a minimális →végrehajtási idő, v. az elfoglalt →operatív memória. Ma már a →fordítóprogramok feladata, a →fordítás fontos fázisa.

kommunikáció (communication) – számítástechnikai értelmezésben rendszerek és/vagy alrendszerek, elemek közötti kapcsolat, amely az információk átáramlását, kicserélését megvalósítja.

kommunikációs csatorna (communication channel) – valamely →adatátviteli hálózat két csomópontja között létrehozott állandó v. időleges kapcsolat, rendszerint bérelt v. saját vonal formájában. A ~ lehet nyilvános v. bérelt telefon-, telex- v. távíróvonal, rádió-összeköttetés stb.

kommunikációs interfész (communication interface) – távolsági adatátvitel megvalósítására használt olyan →interfész, amely kapcsolatot teremt a számítógéprendszer és a →modemek között.

kommunikációs processzor (communication processor) – →adatátviteli hálózatokban kisszámítógép, amely az adatátvitel vezérlésével kapcsolatos feladatokat végzi (vonalak vezérlése, →üzenetek, ill. →adatcsomagok kezelése, kódok átalakítása, hibavédelem, adatkapcsolat-vezérlő eljárások stb.). Ellátja az →adatvégállomás periferikus egységeinek vezérlését is, s ugyanakkor helyi adatfeldolgozási feladatokat vállalhat magára. Alkalmazása jelentősen tehermentesítheti a központi számítógépet, s így az egész hálózat teljesítményét növeli.

kommunikációs terület (communication area) – alkalmazási programok részére az →operatív memóriában programonként kijelölt terület, amely az →adatbázis-kezelő rendszer és az alkalmazási programok közötti információcsere eszköze.

kompaktlemez (compact disk) – optikai módszerekkel letapogatható, modern technológiájú →adathordozó.

kompatibilitás (compatibility) – két berendezés (v. programtermék) együttműködését, közös felhasználhatóságát jellemző tulajdonság. A számítástechnika szempontjából külön vizsgálják az ún. hardver~t és szoftver~t.

Hardver~ áll fenn két számítógép között, ha az egyik →gépi kódját a másik közvetlenül (tehát bármiféle módosítás nélkül) képes végrehajtani (értelmezni). Ez jellemző pl. ugyanazon számítógépcsalád két tagjára (még ha csak egy irányban is), ha az újabb változat a régire írt programokat helyesen hajtja végre. Gyakorlati megvalósítása történhet úgy, hogy az új utasításkészlet magában foglalja a régit is, v. úgy, hogy az új →központi vezérlőegység →emulálás útján értelmezi a régi utasításkészletet.

Ugyancsak hardver~ áll fenn két hasonló célú, de különböző műszaki megoldást választó berendezés között, ha azok minden változtatás nélkül kapcsolhatók ugyanabba a számítógéprendszerbe.

Szoftver~ról beszélnek akkor, ha valamely magas szintű programozási nyelven írott program változtatás nélkül futtatható két különböző számítógéprendszeren. A szoftver~t segíti elő a programozási nyelvek szabványosítása, bár a gyakorlat azt mutatja, hogy a konkrét megvalósítások rendszerint csak alapváltozatként használják a nyelvek standard formáját. Igaz, ilyenkor a ~ gyengülését a magasabb hatékonyság kárpótolja.

kompilátor (compiler) – a →fordítóprogrammal egyenértékű fogalom az angol szaknyelvből. A számítástechnikai szakirodalomban olykor a ~ a fordítóprogramok egyik osztályát képviseli: olyan fordítóprogramra vonatkozik, amelynek →tárgynyelve valamilyen →gépi nyelv (megkülönböztetésül azoktól, amelyeknek tárgynyelve →asszembler nyelv v. a forrásnyelvtől különböző magas szintű programozási nyelv).

komplemens kód (complement code) – a számoknak leggyakrabban használt bináris kódolási módja a numerikus számítógépek-

ben. Ha C(x) jelöli az x szám →fixpontos számábrázolású ~ját, n pozíciós bitvektor felhasználásával a

. pozitív számoknál C(x) = x

. negatív számoknál 1-es ~ban $C(x) = 2^n+x$

. negatív számoknál 2-es ~ban $C(x) = 2^n+x+1$

Az ábrázolható számok tartománya:

. 1-es ~ban: $-2^{n-1} < x < 2^{n-1}$

. 2-es ~ban: $-2^{n-1} \le x < 2^{n-1}$

Pozitív számok 1-es és 2-es ~ja megegyezik egymással és a számok →direkt kódjával. Negatív számoknál a ~ képzési szabályai a következők:

. 1-es ~nál bitenként komplementálnak

. 2-es ~nál az 1-es ~ legkisebb helyértékhez hozzáadódik 1.

Ilyenformán a pozitív számok előjelbitje 0, a negatív számoké pedig 1, függetlenül a használt kódtól. A ~ jelentősége elsősorban az additív műveleteknél jelentkezik, ugyanis ezeket közvetlenül el lehet végezni. Multiplikatív műveleteknél a negatív számok komplementálásával tkp. a direkt kódra vezetik vissza a műveletek elvégzését.

komplex [adat]típus (complex [data] type) – egyes magas szintű programozási nyelvek (pl. →FORTRAN) elemi →[adat]típusa. Két →lebegőpontos számábrázolású adat rendezett párja, ahol az első a komplex adat valós, a második pedig a képzetes része. A ~hoz aritmetikai, összehasonlítási és értékadó műveletek vannak rendelve (ezeket rendszerint →standard függvényekkel oldják meg).

komplex utasításkészletű számítógép (complex instruction set computer) – számítógépek egyik alapvető →architektúrája. Lényege az, hogy az összes elképzelhető funkcióhoz önálló →gépi utasítás tartozik, így →utasításkészlete sokszor jóval meghaladja a 200-at. Előnye, hogy a magas szintű programozási nyelveket szinte egy az egyben lehet fordítani, hátránya viszont, hogy →központi vezérlőegysége igen komplex, tehát igen drága is. (Másik alapvető architektúratípus a →korlátozott utasításkészletű számítógép.)

konfiguráció (configuration) – számítógéprendszer →architektúrájának jellemzője, amely meghatározza a rendszerhez tartozó →hardveregységek számát és paramétereit. Minimális ~n azokat a hardveregységeket értik, amelyek feltétlenül szükségesek bizonyos programtermékek végrehajtásához. A ~ máskülönben alapvető meghatározója a konkrét számítógéprendszernek a →rendszergenerálás szempontjából.

konfigurációs állomány (configuration file) – →operációs rendszerekben használt olyan speciális →[adat]állomány, amely az egyes programtermékeknek konkrét környezetben való végrehajtásához szükséges környezeti paramétereket beállító parancsokat tartalmazza.

konjunkció (conjunction) – →logikai szorzás

konjunktív normálalak (conjunctive normal form) – adott →logikai függvény argumentumában szereplő változók és negáltjaik diszjunkciójának konjunkciója (logikai összegének logikai szorzata), amiből minden összetett feltétel előállítható (→diszjunktív normálalak).

konkatenáció (concatenation) – →jelsorozatokkal végezhető művelet, operandusai: két karakterlánc, a művelet eredménye pedig a két lánc összekapcsolásából keletkezett új karakterlánc. A →PL/1 és más, jelsorozatokkal (is) dolgozó magas szintű programozási nyelv jellegzetes művelete, amelyhez megfelelő utasítások és →standard függvények tartoznak.

konkrét szintaxis (concrete syntax) – adott →programozási nyelv olyan →szintaxisa, amely szimbólumsorozatokból felépített helyes programok előállítására, ill. ellenőrzésére vonatkozik. A →fordítóprogramok, ill. →értelmezőprogramok gyakorlatából kifejlődött fogalom, amely a programok konkrét megjelenési formájának értelmezésére és lefordítására alakult ki, megszüntetve az →absztrakt szintaxis felesleges elemeit, de kibővítve azt a gyakorlati tevékenység végrehajtásához szükséges eszközökkel. Ilyenformán egy absztrakt szintaxishoz több ~ is kapcsolható. A modern fordítóprogramok túlnyomó többsége az adott nyelvekre definiált ~ alapján működik.

konkurens elvű rendszer (concurrent system) – komplex rendszerek speciális szervezési módja. Lényege az, hogy a teljes rendszert részlogikájukban egymástól függetlenül saját szinkronizmusuk szerint működő részrendszerekre bontják, és a belőlük felépített, egy szinttel teljesebb rendszer magasabb absztrakciós szintjén megvalósítják az egymáshoz képest aszinkron módon működő részelemek összehangolt, helyes rendszerét. A ~ fogalma megtalálható a →szoftver területén is: végeredményben ~nek lehet tekinteni a több absztrakciós szintből álló többfolyamatos rendszert, amelynek folyamatai között a kölcsönös függés megvalósításához koordinációra (szinkronizációra) van szükség. Megvalósítására megjelentek a konkurens elvű programozást is támogató magas szintű nyelvek (→konkurens Pascal, →Ada, →CHILL), amelyek speciális nyelvi eszközökkel rendelkeznek a konkurens folyamatok programozására is.

konkurens Pascal (concurrent Pascal) – a →Pascal kibővítéseként előállított olyan magas szintű, strukturált programozási nyelv, amely a nyelvi jelölésrendszert a konkurens programozásra is kiterjeszti. A szokásos programozási nyelvekkel szemben lehetőséget ad konkurens programok, így →operációs rendszerek, valós idejű folyamatirányítási rendszerek, számítógéphálózatok kapcsolási rendszereinek magas szintű nyelven történő leírására is. Különleges jelentősége van a →mikroszámítógépek területén, ahol gyakori feladat célorientált rendszerek gyors és megbízható készítése, e célra a nyelv tömörsége, pontos és egységes fogalomrendszere, modularitása és jó dokumentálhatósága teszi alkalmassá.

konstans (constant) – olyan adat, amelynek értéke változatlan marad a program egész végrehajtási ideje alatt, s ezt az értéket megjelenési formája önmagában definiálja. A magas szintű programozási nyelvekben a ~ok közvetlenül felhasználhatók a →forrásnyelvű szövegben, ugyanis a →fordítóprogramok, ill. az →értelmezőprogramok feladata ezek számára tárterületet biztosítani és értéket generálni. A ~ legismertebb típusai: →alfanumerikus konstans, →figuratív konstans, →logikai konstans, →numerikus konstans, →címkekonstans stb.

kontrollösszeg (checksum) – információt hordozó jelsorozat valamilyen modulo összege, esetleg ennek komplementuma. Az, hogy milyen szám szerint képezik a modulo összeget, attól függ, hogy milyen hosszúságú sorozatokra van bontva a teljes információ (pl. 8 csatornás lyukszalagról történő bevitelnél ez 2^8). A ~ használatának elsőrendű célja az információ átviteli v. rögzítési helyességének ellenőrzése.

konvergencia (convergence) – valamely →iterációs eljárás eredményességének feltétele. Ha valamilyen numerikus értéket iterációs eljárással számítanak ki, és az eljárás olyan, hogy lépésről lépésre a várt eredmény egyre jobb megközelítését adja, akkor az eljárás konvergens. A →numerikus analízis egyik alapvető kérdése a numerikus módszerek ~jának vizsgálata.

konverziós hiba (conversion error) – numerikus adatoknak egyik ábrázolási módból másikba történő átalakítása során esetleg előforduló hiba. Egyike azoknak a legfontosabb hibatípusoknak, amelyeket a →hibaelemzésnek feltétlenül figyelembe kell vennie.

konzolírógép (console typewriter) – a számítógéprendszer és a gépkezelő közötti információcsere hagyományos eszköze, a számítógéppel közvetlen →adatcsatornán kommunikáló elektromos írógép. Korszerűbb kivitelezése a képernyős →vezérlőpult.

korlátozott bejegyzés (limited entry) – egyszerű →döntési táblázatokra jellemző bejegyzés, amely vonatkozhat a feltételtörzsre és a tevékenységtörzsre egyaránt. Azt jelenti, hogy a feltételek állapotainak jelölésére elégséges egy hármas értékkészlet (igaz, hamis, ill. közömbös), a végrehajtandó tevékenységek jelölésére pedig egy kétértékes készlet (az adott tevékenység végrehajtandó-e v. sem).

korlátozott utasításkészletű számítógép (reduced instruction set computer) – számítógépek egyik alapvető →architektúrája. Lényege az, hogy csak néhány, alapjellegű funkcióhoz rendel önálló →gépi utasítást, így →utasításkészlete sokszor alig éri el az 50-et. Előnye, hogy →központi vezérlőegysége igen egyszerű, tehát olcsó is, viszont a magas szintű programozási nyelvek utasításait csak rövidebb-hosszabb gépi utasítássorral lehet

lefordítani, így a →tárgyprogramok is igen hosszúak. (Másik alapvető architektúratípus a →komplex utasításkészletű számítógép.)

korutinok (coroutines) – olyan programegységek, amelyek felváltva játsszák a hívó, ill. hívott eljárás szerepét; egyik sem alárendelt viszonyú a másikkal szemben. Ez a mechanizmus látszólag párhuzamos végrehajtást biztosít: a független programegységek aktív állapota csupán a szinkronizálási feltételektől függ.

A ~ megvalósításához speciális mechanizmus szükséges, amely nyilvántartja a mindenkori pillanatnyi állapotokat, a vezérlésátadás pontjainak helyét és jellemzőit. E mechanizmust rendszerint →szoftverelemekkel valósítják meg, de egyes rendszerekben →hardverlehetőségek is adódhatnak rá. A ~ felhasználása előnyös a →konkurens elvű rendszerek (→valós idejű rendszerek, →operációs rendszerek, →fordítóprogramok stb.) megvalósításában.

kölcsönös kizárás (interlock) – →osztott erőforrást közösen használó, párhuzamosan futó folyamatok megoldhatatlannak látszó versenyhelyzete. ~nál az összes érdekelt folyamat felfüggesztett állapotba kerül, mert ugyanarra a már foglalt erőforrásra tartanak igényt. Megoldásához rendszerint az →operációs rendszer közbelépésére van szükség, s általában a →prioritások újrakiosztásával jut egy-egy folyamat elsőbbségi helyzetbe a többivel szemben.

kölcsönös szinkronizálás (handshaking) – adatátviteli műveletekben használatos kommunikációs módszer. Alapvető jellemzője az, hogy minden egyes jelet egy ellenkező irányban haladó válasz követ. Így az átvitelben részt vevő mindkét oldal befolyásolja a működési sebességet. A tényleges adatátvitelt jól meghatározott sorrendű vezérlőjelváltás előzi meg. Ha az adó vezérlőjelére a vevő nem válaszol az előre meghatározott válaszszal, az adó nem küldi ki a hasznos információt. Az adó a dialógust kezdeményező vezérlőjel kiadása után várakozó állapotba kerül, és abban marad mindaddig, amíg a vevő azt nem jelzi, hogy kész az információk vételére.

kölcsönvétel (borrow) – az a jelzés, amikor a kivonási műveletnél a kisebbítendő számjegy kisebb, mint a kivonandó, s a művelet elvégzéséhez „kölcsön" kell venni a következő pozíció egy egységét (az összeadásnál esetleg fellépő →átvitellel rokon fogalom).

könyvtár (library) – számítástechnikai értelmezésben: programoknak, programegységeknek v. programmoduloknak valamilyen mágneses →adathordozón (rendszerint →mágneslemezen v. →mágnesszalagon) tárolt, rendszerezett halmaza. A ~ra jellemző információkat (a ~ nevét, formáját, létrehozási dátumát, védelmi idejét, a benne foglalt elemek nevét és tárolási sorrendjét stb.) →katalógusnak nevezett táblázatban rögzítik. A ~ak előállítását, törlését, változtatását, javítását, listázását stb. speciális segédprogram , a →könyvtárkezelő program biztosítja. A tárolt egységek – „könyvek" – kódolási formája egyben a ~ típusát is rögzíti: lehet →forrásnyelvű ~, →áthelyezhető bináris formájú ~, közvetlenül végrehajtható formájú ~ stb. Az →operációs rendszerek szempontjából megkülönböztetett jelentőségű a →szubrutinkönyvtár és a →makrokönyvtár.

A ~ a lemezen lévő állományok csoportjait foglalja össze, és tartalmazhat másik ~at, a ~ak a gyökérkönyvtárakból kiindulva faszerűen épülnek fel.

könyvtárkezelő program (librarian) – →operációs rendszerekhez tartozó →segédprogram, amely a →könyvtárak kezelésével, karbantartásával kapcsolatos műveletek elvégzését biztosítja. Ezek a műveletek az előállítás, az aktualizálás, a javítás, a törlés, az átmásolás, a tartalomkivonatolás stb., valamint a megfelelő →katalógus alapján végezhetők el. A felhasználók igényeit specializált parancsnyelv utasításai alapján értelmezi és szolgálja ki a ~.

körkörös léptetés (circular shift) – →ciklikus léptetés

körkörös lista (circular list) – olyan →egyirányú lista, amelynél az utolsó adatelem mutatója az adatszerkezet első elemére mutat. Az ilyen adatszerkezetnél valamely adatelem lokalizálása bármelyik felépítő elemből kiindulva megvalósítható.

környezetfüggetlen nyelv (context-free language) – →környezetfüggetlen nyelvtan által generált programozási nyelv (→Chomsky-hierarchia, →generatív nyelvtan).

környezetfüggetlen nyelvtan (context-free grammar) – a →Chomsky-hierarchia szerinti osztályozás alapján a harmadik osztályba tartozó →generatív nyelvtan. Alapvető jellemvonása: ha a →helyettesítési szabály bal oldala egyetlen →nemterminális szimbólumból áll, a szabály alkalmazása nem függhet a nyelvtani változó környezetétől. A programozási nyelvek elmélete és gyakorlata szempontjából a ~nak van a legnagyobb jelentősége, ugyanis a programozási nyelvek →szintaxisát (az →ALGOL 60-nal kezdve) ~ segítségével írták le. Bár ez a leírás formailag különbözik a →Backus-Naur Formától, lényegileg ugyanazon az elméleten alapszik.

környezetfüggő nyelv (context-sensitive language) – →környezetfüggő nyelvtan által generált nyelv (→Chomsky-hierarchia, →generatív nyelvtan).

környezetfüggő nyelvtan (context-sensitive grammar) – a →Chomsky-hierarchia szerinti osztályozás alapján a második osztályba tartozó →generatív nyelvtan. A →helyettesítési szabályokra vonatkozó megszorítás kiköti, hogy minden szabály jobb oldalán legalább annyi szimbólum szerepeljen, mint a bal oldalon. A ~ok alkotják azt a legszélesebb nyelvtanosztályt, amelyben a nyelvhez tartozás kérdése minden sorozatra véges időn belül általános algoritmussal eldönthető. A →programozási nyelvek leírásának eszközeként mégsem terjedt el, ugyanis a környezetfüggőség előnyeinek kihasználása igen nagy terjedelmű nyelvtanokat eredményezne.

kötegelt feldolgozás (batch processing) – 1. Számítógéprendszer olyan alapvető üzemeltetési módja, amelynél az elvégzendő munkák előre rögzített feladatkötegek formájában érkeznek a számítóközpontba, s a munkák elvégzésének idején a számítógép és a felhasználó között rendszerint nincs közvetlen kapcsolat. Mivel ~nál egy-egy feladat végrehajtására nem egy konkrét időpont, hanem napos nagyságrendű időintervallum áll rendelkezésre, az →operációs rendszernek (s egyben a gépkezelőnek) lehetősége van a munkák olyan csoportosítására, ami biztosíthatja a →hardver és →szoftver →erőforrások optimális kihasználását. A munkák végrehajtásának sorrendjét a gépkezelő v. maga az operációs rendszer határozza meg, a konkrét felhasználói követelmények és a saját erőforrások gazdaságos kihasználását biztosító feltételek alapján. (A másik alapvető üzemeltetési mód a →párbeszédes üzemmód.) – 2. A batch programok az MS-DOS rendszerekben, a COM a VAX rendszerekben olyan szöveges állományok, amelyek több programindítást v. egyéb műveletet hívnak meg adott sorrendben.

kötegelt feldolgozású terminál (batch processing terminal) – olyan →terminál, amely saját vezérlőegységgel és a kapcsolt központi számítógépéhez hasonló →periferikus egységekkel rendelkezik. Jelentősége, hogy a felhasználó a számítógéptől távol is elvégezheti mindazokat a műveleteket, amelyeket másküIönben a számítóközpontban kellene elvégeznie (munkák és a feldolgozáshoz szükséges bemenő adatok rendelkezésre bocsátása végrehajtás céljából, ill. az eredmények átvétele). A ~ minimális →konfigurációja legalább egy kártyaolvasót (v. más adatbeviteli eszközt) és egy nyomtatót tartalmaz, de ha vezérlőegysége megengedi, kiterjeszthető más periferikus egységekkel is. Sokszor saját (korlátozottabb) feldolgozási lehetőségekkel is rendelkezhet (→adatvégállomás, →adatállomás).

kötet (volume) – mágneses →adathordozók (→mágnesszalag, →mágneslemez) olyan fizikai egysége, amely időlegesen kapcsolható a számítógéprendszerhez. ~ pl. egy mágnesszalagtekercs, egy mágneslemezcsomag stb. Szabványos szervezésnél a ~et →adathordozói címkékkel látják el, amelyek a ~ azonosítására szolgáló információkat tartalmazzák (→automatikus kötetfelismerés).

kötet-zárócímke (end-of volume label) – →mágnesszalagoknál használt →szabványos címke az →adathordozó logikai végének jelzésére. Elsősorban a több tekercset elfoglaló →[adat]állományok kezelésénél játszik szerepet.

következő függvény (successor function) – a →sorszámozott [adat]típust használó programozási nyelvekben (pl. →Pascal, →Ada, →CHILL stb.) olyan standard függvény, amely az →aktuális paraméterként megadott szimbólumhoz az eggyel nagyobb sorszámú következő szimbólumot rendeli

hozzá. Pl. ha a következő deklarációt használták:

var Nyelv: (magyar, német, angol)

akkor a ~: **succ** (német) értéke angol lesz.

következtető gép (inference engine) – →ismereten alapuló rendszerek alapvető része, amely általános logikai-következtetési módszerekkel dolgozza fel az →ismeretbázisban megfelelően ábrázolt →tárgyköri tudást, s ezáltal új ismeretekkel, következményekkel gazdagítja az eredeti ismeretbázist.

közbülső nyelv (intermediate language) – programok →fordítása közben létrejövő átmeneti formák jelölésére, kódolására használatos nyelv. Kevésbé komplex, mint a →forrásnyelv, de magasabb szintű a →tárgynyelvnél. Megtörténhetik, hogy a ~ adott számítógép →asszembler nyelve, s ilyenkor a végleges tárgynyelvű programot az →asszembler állítja elő. →Gépi fordításnál is használhatnak ~et, érdekes példa erre az egyes rendszerekben ~ként használt eszperantó nyelv.

közös szegmens (common segment) – speciális →adatszegmens, amely több →programszegmens által is elérhető, közösen használható adatokat tartalmaz (→átfedési szerkezet, →szegmentálás).

központi egység (central unit) – a →központi vezérlőegységből és az →operatív memóriából álló együttes elnevezése.

központi memória (central memory) – →operatív memória

központi monitor (central monitor) – 1. Az →operációs rendszerek magját alkotó legfontosabb vezérlőprogramok alrendszere, a →monitornak állandó jelleggel az →operatív memóriában található része. Legfontosabb feladata a →hardver és az operációs rendszer többi részei közötti kapcsolat megteremtése, vagyis a →kiterjesztett gép utasításainak végrehajtása. Legfontosabb része a →felügyelőprogram (szupervizor), amelynek fő feladata a →multiprogramozás megvalósítása, a →megszakítási rendszer vezérlése, valamint a fizikai adatcsere ellenőrzése. A ~ más fontos funkciói: a →szegmensek betöltése és elindítása, programok közötti információcsere megvalósítása, programok végrehajtásának befejezése, →feladatok (taszkok) létrehozása, aktivizálása és megszüntetése, belső órák kezelése, a gépkezelővel folytatott dialógus biztosítása stb. – 2. Hálózatok esetén (pl. Novell) a rendszer pillanatnyi állapotát mutató program.

központi terminál (central terminal) – →párbeszédes üzemmódban dolgozó rendszerekben az a speciális funkciókat ellátó (rendszerint képernyős) →hardveregység, amelynek segítségével megvalósul egyrészt az →operációs rendszerrel folytatott dialógus, másrészt figyelemmel kísérhető és vezérelhető a rendszerre kapcsolt többi (felhasználói) terminál működése. Ez utóbbi funkció keretében a ~ üzeneteket fogadhat a felhasználói termináloktól, →prioritási, ill. jogosultsági szinteket oszthat ki, kérésre csak az ő hatáskörébe tartozó műveleteket végezhet stb.

központi vezérlőegység (central processing unit) – a számítógéprendszer egyik alapvető funkcionális egysége, az egész rendszer vezérlője, amely az utasítások dekódolását elvégzi, és ennek alapján a számítógép egyéb részeinek, ill. a csatlakozó →perifériának a vezérlését biztosítja. Meghatározó funkcionális jellemzői: az →utasításkészlet, a →gépi kód, a →címzési módszerek, az adatábrázolási módszerek, a →műveleti sebesség stb.

közvetett cím (indirect address) – →indirekt cím

közvetett hozzáférés (indirect access) – címezhető →adathordozókon (pl. →mágneslemezen) rögzített →[adat]állományok bejegyzéseinek olyan hozzáférhetősége, amelynél az egyes →rekordok címeit az [adat]állományban tárolt és előzőleg kiolvasott bejegyzésekből nyerik. ~sel olvashatók ki a bejegyzések az olyan →szekvenciális [adat]állományból, amelyben a logikai sorrend és a fizikai tárolási sorrend nem egyezik meg, hanem a logikai sorrendet →mutatók segítségével megvalósított láncolás biztosítja. Ugyancsak a ~ teszi lehetővé az →indexszekvenciális [adat]állományok bejegyzéseinek elérését is, itt a címezhetőséget a →sávindexek, ill. →cilinderindexek előzetes beolvasása biztosítja.

közvetlen hibajavítás (forward error correction) – →adatátviteli hálózatokban használatos →hibakorlátozási stratégia, amely hatékony →hibajavító kódot igényel. A hibakorlátozó dekódoló igen bonyolult és költséges

(főleg ha erősen redundáns hibajavító kódot használnak), viszont ez a stratégia nem igényel visszacsatoló csatornát. Nagy sebességű egyirányú összeköttetéseknél alkalmazása gazdaságos lehet.

közvetlen hozzáférési módszer (direct access method) – kizárólag címezhető →adathordozókon (rendszerint →mágneslemezen) tárolt adatok elérésére használt módszer. Lényege az, hogy a →logikai rekordok kulcsa és fizikai címe között egyértelmű a megfeleltetés. Legegyszerűbb a megvalósítása, ha a kulcsok numerikusak, a →blokkok és a rekordok rögzített hosszúságúak. Ilyenkor a fizikai cím meghatározásához egyszerű osztási művelet elegendő. Ez a módszer azonban elég ritkán használható, v. azért, mert a kulcsok nem numerikusak, v. pedig azért, mert a kulcsok lehetséges értékei sokkal nagyobb tartományt ölelhetnek fel, mint ahány rekord az [adat]állományban ténylegesen előfordulhat. Gyakorlatilag a megfelelően választott leképzési algoritmusok (→terítőfüggvények) biztosítják a fizikai címek meghatározását.

közvetlen hozzáférésű [adat]állomány (direct access file) – →direkt szervezésű [adat]állomány

közvetlen hozzáférésű adathordozó (random access media) – olyan →adathordozó, amelyen minden →fizikai bejegyzésnek külön- külön egyértelmű címe van. Ennek alapján az illető bejegyzés közvetlenül elérhető anélkül, hogy a megelőző bejegyzéseket végig kellene olvasni. A legfontosabb ~k: a →mágneskártya, a →mágneslemez stb.

közvetlen operandus (immediate operand) – operandus, amely nem a →címével szerepel a →gépi utasításban, hanem magával az értékével van jelen az utasításszóban. A →címrész tartalmának valódi jellegét az utasítás kódja dönti el: az utasításkód felismerése után a →központi vezérlőegység értelmezi a címrészt mint ~t v. mint operanduscímet.

közvetlen tárhozzáférés (direct memory access) – speciális adatátviteli eljárás, amely lehetővé teszi adatok közvetlen cseréjét az →operatív memória és a →periféria között anélkül, hogy a →központi vezérlőegység beavatkozását igénybe venné. A ~ a →cikluslopás elvén alapszik, s a nagy sebességű átviteleknél alkalmazzák (→közvetlen tárhozzáférési egység).

közvetlen tárhozzáférési egység (direct memory access unit) – az →operatív memóriát a →központi vezérlőegységtől függetlenül címezni képes, nagy sebességű adatátvitelt biztosító hardverberendezés, amely 1 szó/memóriaciklus sebességet tud megvalósítani anélkül, hogy a központi vezérlőegységet lefoglalná. Saját nagy sebességű vezérlőegységének működési elve a következő: a →puffermemória kezdőcímét a ~ címregiszterébe tölti; adatbevitelt feltételezve, az első adatot a ~ munkaregiszterébe küldi; a →b/k gyűjtősínen keresztül speciális megszakítást kérve, az adatot a következő memóriaciklussal az operatív memóriába küldi. A saját vezérlőegység parancsok formájában kapja a működéséhez szükséges vezérlőjeleket, ugyancsak a b/k gyűjtősínen keresztül.

közvetlenül végrehajtható forma (executable form) – →abszolút bináris forma

kritikus erőforrás (critical resource) – olyan →erőforrás, amelyet egyszerre csak egyetlen folyamat használhat. Az ilyen típusú erőforrást is lehet →osztott erőforrásként használni, ha bizonyos vezérlési feltételek adottak (→kritikus szakasz, →szemafor).

kritikus szakasz (critical section) – párhuzamosan futó konkurens folyamatok olyan részei, amelyek adott →erőforrást közösen használva kölcsönösen kizárják egymást, vagyis egy időben az illető erőforrást kizárólag egyetlen folyamat használhatja. A holtpontmentességet (→kölcsönös kizárás) a következő feltételek biztosíthatják: az osztott erőforrásra vonatkozóan a ~ban egyidejűleg legfeljebb egy folyamat tartózkodjék; bármelyik folyamat beléphet a ~ba akkor, ha egyetlen más folyamat sem tartózkodik már ebben az állapotban.

kulcs (key) – a →heterogén adatstruktúrákat felépítő, kitüntetett szereppel rendelkező adatmezőhöz rendelt érték, amelynek alapvető szerepe az, hogy segítségével azonosíthassák az adatelemeket. Két legfontosabb típusa, amelyek a gyakorlatban majdnem fedik egymást, a →besorolási kulcs és a →keresési kulcs.

kulcsszó (keyword) – többnyire olyan →lefoglalt szó, amelynek előre definiált jelentése van a programozási nyelv →szintaxisában. ~

pl. a →COBOL nyelvben az ADD, a DIVIDE, a COMPUTE stb.

kurzor (cursor) – →helyőr

külön fordítás (separate compiling) – korszerű programozási módszer, amely lehetővé teszi nagy, komplex programok részekre tördelését és ezek önálló →fordítását. A módszer egy kezdetleges változata (független fordítás) már a klasszikus nyelveknél is ismert volt (pl. →FORTRAN, →COBOL, →PL/1 stb.), a két módszer azonban alapvetően különbözik egymástól. A független fordításnál a →fordítóprogram nem vesz figyelembe semmilyen forrásnyelvi kapcsolatot a különböző részegységek között, a →külső hivatkozások feloldását a →kapcsolóprogramra bízza. Az →erősen típusos nyelvek megjelenésével a fordítóprogram – amelyre igen komoly ellenőrzési feladatok hárulnak (→típusegyezés) – csak úgy tehet eleget feladatainak, ha már fordításnál ismeri a különböző egységek közötti kapcsolatok attribútumait. A módszer éppen ezen kapcsolatok formális (forrásnyelvi szintű) leírására, ill. feldolgozására vonatkozik.

külső adat (external data) – olyan →adat, amely a feldolgozás pillanatában valamelyik külső →adathordozón található. Ez v. még feldolgozásra vár, v. a program állította elő és küldte ki az adathordozóra. Alapvető jellemzője, hogy a →központi vezérlőegység által közvetlenül nem címezhető, tehát nem is dolgozható fel, csak akkor, ha előzőleg beviteli utasítás eredményeként az operatív memóriába kerül (→belső adat).

külső hivatkozás (external reference) – hozzáférés valamely programegységben definiált objektumhoz más programegységből, az objektum szimbolikus nevének segítségével (→globális szimbólum). A →moduláris programozás megvalósíthatóságának alapvető eszköze. A szimbolikus hivatkozások a program →kapcsolásakor, esetleg a →betöltésnél oldódnak fel.

külső megszakítás (external interruption) – olyan →megszakítás, amely lehetőséget ad a →központi vezérlőegységnek arra, hogy külső jelekre, eseményekre, ill. hardverórától jövő jelekre reagáljon, válaszoljon. ~ok segítségével valósulhat meg a →valós idejű →folyamatirányítás, az aszinkron módon működő →adatcsatornák, ill. →periferikus művek felügyelete stb. →Kötegelt feldolgozással működő rendszereknél igen fontos a →gépkezelői beavatkozást megvalósító ~.

külső memória (external memory) – →háttértár

külső rendezés (external sorting) – nagy adathalmazok →rendezésére alkalmazott módszer, amely részműveleteiben külső →adathordozókat használ. Lényege az, hogy az →operatív memóriában valamilyen →belső rendezési módszerrel rendezett részhalmazokat állítanak elő, majd ezeket kiírják valamilyen perifériára, rendszerint →mágnesszalagra v. →mágneslemezre. A következő lépés(ek)ben ezeket →összeválogatással rendezik egyre nagyobb részhalmazokba. Attól függően, hogy milyen módon viszik ki a rendezett részhalmazokat, majd utána hogyan egyesítik őket, különböző ~i módszerek ismeretesek, közülük a legfontosabbak az →egyenletes összeválogatás, az →oszcilláló összeválogatás és a →többfázisú összeválogatás.

kvadratikus rendezés (quadratic sorting) – az →összeválogatásos rendezés speciális módszere, amelynél a részhalmazok nagysága kb. egyforma, számuk pedig kb. a rendezendő elemek számának négyzetgyökével egyenlő.

kvantálási hiba (quantization error) – analóg mennyiségek digitális jelekké alakítása közben előforduló hibatípus, az →analóg-digitális konverterek jellegzetes hibája.

kvázipárhuzamos végrehajtás (quasiparallel execution) – egyetlen →processzorral rendelkező, több felhasználót kiszolgáló rendszerekre jellemző működés, programvégrehajtás. Lényege az, hogy a rendszer →erőforrásainak működési sebessége és a felhasználók reakcióideje között fennálló (sokszor több nagyságrendű) különbség következtében a rendszer látszólag egy időben több felhasználót is képes kielégíteni. Jellegzetes képviselője az →időosztásos rendszer. A ténylegesen párhuzamos végrehajtást természetesen csak a két v. több processzorral rendelkező rendszerek biztosíthatják.

L

láncolás (chaining) – különböző →adatstruktúráknak az →operatív memóriában v. a címezhető →háttértárban történő ábrázolására kifejlesztett módszer. Lényege, hogy a fizikai tárolási sorrendtől függetlenül a →mutatók segítségével biztosítható a felépítő elemek logikai sorrendje, ill. a logikai sorrendben történő feldolgozása. Ez utóbbi azt jelenti, hogy a pillanatnyilag feldolgozott adat – a feldolgozandó információk mellett – a következő feldolgozandó adat címét is biztosítja.

láncolt [adat]állomány (linked file) – a →fizikai [adat]állománynak csak címezhető →adathordozón (pl. →mágneslemezen) megvalósítható szervezési formája. Lényege, hogy az adathordozó egy összefüggő területét több egyenlő nagyságú zónára osztják, s ezeket utalják ki egy v. több azonos típusú [adat]állomány részére. A kiosztás a tényleges igényeknek megfelelő sorrendben történik, és az ugyanazon [adat]állományhoz tartozó zónákat belső mutatók fűzik össze (a zónák elhelyezkedése nem feltétlenül folytonos), igen gazdaságos helykihasználást biztosítva.

láncolt kód (threaded code) – →programozási módszer, amelyet elsősorban egyes →fordítóprogramoknál használnak. Lényege, hogy az előállított →program, ill. →tárgyprogram kizárólag →szubrutinokat hívó utasításokból áll.

lap (page) – 1. →Lapkezeléssel megvalósított dinamikus memóriakiosztásnál az →operatív memória kiutalási, ill. felszabadítási egysége, a programoknak pedig betöltési egysége. A ~ok nagyságának megválasztása jelentősen befolyásolhatja a lapkezelés hatékonyságát. Ha túl nagyok a ~ok, a memória kihasználása igen leromolhat, mivel sok hely marad kihasználatlanul. Ha viszont túl kicsik a ~ok, a laphelyettesítéssel és az adminisztrációs információk felduzzadásával a betöltéssel kapcsolatos műveletek időigénye nőhet a megengedett határokon túlra. A gyakorlatban használatos bevált értékek általában 1024, 2048, ill. 4096 →bájt. – 2. Mikrogépeknél az egy utasítással közvetlenül címezhető egység, általában 256 bájt (→lapozott címzés).

lapcséplés (thrashing) – a kétszintes →virtuális memóriánál és a →lapkezelésnél leg-több gondot okozó jelenség, amely abban nyilvánul meg, hogy a nem megfelelő →laphelyettesítési algoritmus miatt a szükséges lapokat ismételten eltávolítják az →operatív memóriából, majd újra beolvassák őket. A túl sok lapozással felgyűlő, improduktív tevékenység teljesítménycsökkenéshez vezet. Egyes, korábban megfelelően viselkedő rendszerekben is felléphet hirtelen, a terhelés növekedésével párhuzamosan, s ezzel az addig hatékony számítógép teljesítménye nagyon leromolhat. Az →operációs rendszer időszakonként végzett kiértékelések útján vehet tudomást a ~ről, és rendszerint a laphelyettesítési algoritmus megváltoztatásával javíthat a helyzeten.

laphelyettesítési algoritmus (page replacement algorithm) – →lapkezeléssel megvalósított →multiprogramozásnál használatos algoritmus, amelynek célja annak eldöntése, hogy memóriaigénylésnél az újonnan behozandó →lap melyik bennlévőt írja felül. Az elméletileg tökéletes ~ azt a lapot választaná, amelyre a leghosszabb ideig egyik program sem fog hivatkozni. Mivel ez gyakorlatilag nem valósítható meg, olyan eljárásokat használnak, amelyek az →operatív memóriában lévő lapoknak, ill. programoknak múltbéli és jelenlegi viselkedéséből indulnak ki. A leggyakrabban használt ~ok:

1) Véletlenszerű lapváltásnál minden ellenőrzés nélkül, találomra kiválasztott lapokat írnak felül. Könnyen megvalósítható módszer, viszont meglehetősen káros, mivel gyakran felülírhat éppen használható lapokat is.

2) „Első-be – első-ki" (FIFO) típusú lapváltásnál az operatív memóriában legrégebben betöltött lapokat áldozzák fel. Értelmezése egyszerű, azonban nem szükségképpen hatékony is, mivel az elsőként beolvasott lap könynyen lehet éppen olyan, amelyikre többször is szükség lehet, tehát nem kellene eltávolítani.

3) A legrégebben használt lap helyettesítése (LRU – Least Recently Used) azon alapszik, hogy ha egy lapra már hosszabb ideje nem hivatkoztak, akkor igen valószínű, hogy még számottevő ideig nem is fognak. Általában elég hatékony, de megvalósítása bonyolult, mivel speciális bejegyzéseket kell rendszeresíteni annak jelzésére, hogy az egyes lapokat mikor használták utoljára.

4) A legritkábban használt lap váltása (LFU – Least Frequently Used) sokszor, de nem mindig hatékonyabb a legrégebben használt lap váltásánál, megvalósítása pedig éppen olyan bonyolult.

5) A változatlan tartalmú lap felülírása abból a meggondolásból, hogy az ilyen lapokat nem szükséges kimenteni a háttértárba a felülírás előtt (eredeti állapotában mindig megtalálható).

6) →Munkahalmaz szerinti lapváltásnál mindig csak azokat a lapokat helyettesítik, amelyek nem tagjai az aktuális munkahalmaznak, ha pedig a rendelkezésre álló memória nem lenne elégséges az egész aktuális munkahalmaz számára, az egész programot háttértárban tárolják egy későbbi aktivizálás számára.

7) Az ATLAS ciklusfelismerő módszert először a FERRANTI-ATLAS számítógépen használták, ez a programok lapmozgatását a ciklusok felismerésével próbálja minimalizálni. Természetesen adatok lapozására nem alkalmas.

8) A tétlenségi (új lap behozására való várakozási) idő csökkentését célzó eljárások az „áldozatot" igyekeznek előre meghatározni és kimenteni, így biztosítva előre helyet a szükséges új lapoknak. Így a tétlenségi idő csak a lapok betöltésére korlátozódik.

laphiba (page fault) – a →lapkezeléssel kapcsolatban az a különleges helyzet, amikor az →operatív memóriából éppen hiányzó egyik →lapra történik hivatkozás. A ~ →megszakítást okoz, ami a program végrehajtásának felfüggesztéséhez vezet, s az operációs rendszernek gondoskodnia kell a hiányzó lap betöltéséről.

lapkezelés (paging) – a →multiprogramozás egyik legfontosabb módszere, amely az →operatív memóriának mint →erőforrásnak megosztott használatára vonatkozik. Lényege az, hogy a végrehajtandó programot fix (egyes esetekben változó) hosszúságú →lapokra osztják, s ezeket csak a tényleges végrehajtás idejére töltik be a memóriába. Mivel az egyes programokhoz tartozó lapok az operatív memóriában szétszórva helyezkednek el, a logikai címeket felhasználás előtt tényleges címekké kell alakítani. Az átalakítás rendszerint →laptáblák segítségével történik: a tényleges címet a hivatkozott lap abszolút betöltési címének és a hivatkozás lapon belüli relatív címének összege adja. A ~ alapvető jelentősége, hogy lehetővé teszi a programok végrehajtását sokkal kisebb operatív memóriában, mint amekkorát a teljes program betöltése követelne (→laphiba, →laphelyettesítési algoritmus).

lapozott címzés (paged addressing) – mikroszámítógépeknél használatos címzési módszer. Az →operatív memóriát 256 bájtos részekre osztják, s ezeken belül 8 →bit elégséges minden cím elérésére. A címzett lap v. az utasítást is tartalmazó lap, v. pedig egy olyan lap, amelynek címe (száma) egy speciális lapregiszterben van.

laprögzítés (page fixing) – →lapkezeléssel dolgozó →operációs rendszerek speciális művelete, amelynek révén bizonyos →lapokat kivon a →laphelyettesítési algoritmusok felügyelete alól. Erre a műveletre különösen akkor van szükség, amikor maga a →felügyelőprogram is lapozható formában van jelen az operatív memóriában.

laptábla (page table) – a →lapkezeléssel megvalósított →virtuális memória →címátalakításában használt járulékos adattáblázat. Minden programhoz egy ~ tartozik, amely a →lapok sorszámát (azonosítóját) és a lapoknak az →operatív memóriában elfoglalt helyét tartalmazza.

lassú memória (slow memory) – a →soros memória jellemzésére használt kifejezés a →közvetlen hozzáférésű memóriától való megkülönböztetés céljából.

láthatósági tartomány (visibility scope) – →erősen típusos nyelvekben használt objektumok alapvető tulajdonsága. →Deklarációjuk révén minden objektumhoz egyértelműen név (→azonosító) és →[adat]típus kapcsolódik. Mindazok a felhasználási pontok, ahol a név és a típus az eredeti deklarációnak megfelelően kapcsolódik, az adott objektum ~át alkotják. Egy objektum ~a rendszerint az a →programblokk, amelyben deklarációja található, kivéve azokat a beskatulyázott blokkokat, ahol ugyanahhoz a névhez – újabb deklaráció révén – más tulajdonságokat kapcsolnak.

lebegőpontos aritmetikai egység (floating-point arithmetic unit) – →lebegőpontos számábrázolású operandusokon aritmetikai

műveletek végrehajtására alkalmas →hardveregység (→aritmetikai egység). A lebegőpontos műveleteket programozott úton is el lehet ugyan végezni, de a ~ jelenléte nagyobb végrehajtási sebességet biztosít.

lebegőpontos művelet (floating-point operation) – →lebegőpontos számábrázolású operandusokkal végzett →aritmetikai művelet v. →normalizálás.

lebegőpontos számábrázolás (floating-point representation) – számok olyan ábrázolása, amikor a

$$\pm m \cdot a^{\pm c}$$

alaknak megfelelően két összetartozó jelsorozat különböztethető meg. Az m →mantissza és a c →karakterisztika kódolási módja és elhelyezése a kódszón belül az adott gépre jellemző és állandó. Az a alap értéke 10 a decimális ábrázolásnál, 2 a binárisnál s leggyakrabban 16 a hexadecimális számábrázolásnál.

lebegőpontos szubrutin (floating-point subrutine) – csak →fixpontos utasítások végrehajtására képes →központi vezérlőegységgel rendelkező számítógépekhez készített speciális programegység, amely →szoftverúton, fixpontos műveletek segítségével szimulálja a →lebegőpontos műveleteket. A csak fixpontos →aritmetikai egységgel rendelkező számítógépekben a lebegőpontos műveletek kódjai ismeretlen utasításokat jelentenek, végrehajtásuk →megszakítást eredményez. A →felügyelőprogram úgy dolgozza föl ezeket a megszakításokat, hogy az utasításkódnak megfelelő ~ra adja a vezérlést.

lefoglalt szó (reserved word) – olyan betűkombináció (rendszerint jelentéssel bíró angol szó v. rövidítés), amely meghatározó szerepet játszik valamely magas szintű →programozási nyelvben. Meghatározott →szemantikai jelentéssel bíró szintaktikai egység, ezért mint →azonosító nem használható. ~ pl. a →COBOL programozási nyelvben a PERFORM, ELSE, IF, LOW-VALUE stb.

leíró modell (descriptive model) – olyan →modell, amely csupán a jelenségek természetét írja le, de nem foglalkozik az okozati összefüggésekkel (szemben a →magyarázó modellel).

leképzés (parsing) – →szintaktikai analízisben a leképzési lépések olyan sorozata, amelynek révén a megfelelő →helyettesítési szabályok alkalmazásával a →kezdőszimbólumtól eljutnak az adott →nyelvhez tartozó bizonyos →mondatig.

leképzési fa (parse tree) – →szintaxisfa

lekérdezés (inquiry) – →adatbázisok által szolgáltatott adatfeldolgozási módszer. A felhasználó – megfelelő azonosítás és minősítés alapján – meghatározott adatait visszakeresheti az adatbázisból. A szolgáltatott információ lehet származtatott adat is, amely nem szerepel fizikailag az adatbázisban, viszont az abban tárolt adatok és összefüggések alapján előállítható.

lekérdezéses szervezés (polling) – →felfűzéses hálózatokban a vonalhasználat megszervezésének alapvető módszere. A vezérállomás – tekintet nélkül a tényleges igényekre – időről időre lekérdezi az összes felfűzött →adatállomást, hogy van-e közölni való üzenete. A lekérdezési sorrendet lista határozza meg.

Láncolt lekérdezésnél a vezérállomás csupán a legtávolabbi adatállomást kérdezi le közvetlenül, majd ez – ha nincs üzennivalója, v. üzenetének végére ért – továbbítja a lekérdezést a vezérállomáshoz.

Névsor szerinti lekérdezésnél a vezérállomás szelektíven hívhatja a tárolt lista alapján párhuzamos vonalakra kapcsolt bármelyik adatállomást. A hívójelet valamennyi adatállomás kiértékeli, de csak a címzett válaszol.

lekérdező nyelv (query language) – →párbeszédes üzemmódban dolgozó rendszerekben az →[adat]állományok, ill. →adatbázisok lekérdezését, más szóval az azokban tárolt adatelemekhez való szelektív hozzáférést támogató →adatkezelő nyelv.

lemezes operációs rendszer (disk operating system) – →operációs rendszerek olyan típusa, amelynél a nemrezidens részt valamilyen →közvetlen hozzáférésű adathordozón (rendszerint →mágneslemezen, kisebb rendszereknél →hajlékonylemezen) tárolják, s innen a különböző, ritkábban használt funkciókat megvalósító →szubrutinokat v. egyes rendszerprogramokat (pl. →fordítóprogramokat, →kapcsolóprogramokat, →betöltőprogramokat, →segédprogramokat stb.) csak a tényleges felhasználás előtt hívják az →operatív memóriába. A rendszerprogramok végrehajtásához szükséges tárolókapacitás

(pl. a fordítóprogramok közbülső kódjai számára) szintén közvetlen hozzáférésű adathordozót igényel.

lengyel forma (polish form) – aritmetikai kifejezések felírásának →postfix és →prefix formájára használatos elnevezés, ugyanis J. Lukasiewicz lengyel matematikus vezette be. Számítástechnikai szempontból több gyakorlati előnnyel rendelkezik: a műveleti jelek olyan sorrendben követik egymást, ahogy végre kell őket hajtani; a műveleti jel közvetlenül megelőzi (v. követi) azokat az operandusokat, amelyekkel az illető műveletet el kell végezni, s így →verem típusú szervezéssel, a forrásszöveg balról jobbra történő folyamatos olvasásával igen egyszerűen ki lehet értékelni az aritmetikai műveleteket, kifejezéseket; a ~ teljesen zárójelmentes, így a →szintaktikai analízisben a műveletek precedenciáját könnyen lehet követni.

lépésenkénti finomítás elve (stepwise refinement principle) – N. Wirth nevéhez fűződő programfejlesztési módszertan, ő rendszert alkotó leírást adott a programok készítésének olyan módszerére, amelyet sok programozó már előzőleg intuitív módon használt. A ~ tkp. nem kizárólagosan a programtermékek előállításának módszere, mert hatékonyan alkalmazható más jellegű, komplex rendszerek tervezésénél is.

Lényege az, hogy a megoldandó feladatot több, kevésbé komplex részfeladatra bontják úgy, hogy azok megfeleljenek a következő kritériumoknak: legyenek önmagukban megoldhatók; bizonyos részfeladat a lehető legkevesebb kihatással legyen a többi megoldására; az összes részfeladat megoldása után az eredeti feladat megoldása csupán a megfelelő illesztések megvalósítására korlátozódjék.

A ~ rekurzív módon alkalmazható, ugyanis az első lebontás részfeladataira újra használható.

Ha P_0 jelöli az eredeti feladatot, a ~ tkp. egy sor olyan P_i részfeladatra bontja azt, hogy

$$P_0 \rightarrow P_1 \rightarrow P_2 \rightarrow \ldots P_i \rightarrow \ldots P_n$$

ahol P_{i+1} a P_i lépésből következik úgy, hogy annak bizonyos részegységeit az adott finomítási lépésben megoldották. A P_1, P_2, P_3, ..., P_n tkp. ugyanannak a feladatnak egyre konkrétabb megoldásait jelentik. A folyamat akkor adja a végső megoldást, ha a kapott algoritmus minden egyes művelete végrehajtható valamely gép segítségével. Ez általában egybeesik valamely →programozási nyelvvel.

A ~ alapvető, kiinduló eleme volt az E. W. Dijkstra által meghatározott, →strukturált programozás néven elterjedt programfejlesztési módszertannak.

lépésenkénti végrehajtás (step-by-step execution) – programok belövési (tesztelési) fázisában a kódolási, ill. a logikai hibák felderítésére használatos módszer, amelynek célja a nagy sebességű végrehajtás lelassítása az emberi kiértékelhetőség szintjére. Lényege, hogy minden utasításciklus, esetleg →gépi utasítás után leállítják a felhasználói rendszer működését, s így azt az ember észlelési és vizsgálati sebességéhez igazítják. A ~ – felhasználói szinten – elsősorban a →párbeszédes üzemmódban dolgozó rendszereknél használható (→ellenőrzött végrehajtás).

léptékprobléma (scaling problem) – →valós idejű rendszerek megvalósításában felmerülő nehézségek összefoglaló elnevezése. Rendszerint akkor lép fel, amikor egy feladat kényszerűen (lineáris megközelítések, numerikus módszerek stb. folytán) leegyszerűsített változatára kidolgozott programtermék a valós világ komplex környezetébe kerülve nem felel meg az elvárásoknak.

léptetés (shifting) – valamely bitvektor olyan értelmű megváltoztatása, hogy minden bitpozíció a szomszédos bitpozíció tartalmát veszi fel. A bitvektor pozícióit balról jobbra számozva, balra ~nél az i rangú bit az i–1 rangú bit helyére, jobbra ~nél pedig az i+1 rangú bit helyére lép. Attól függően, hogy a bitvektor szélső bitpozíciói milyen módon vesznek részt a műveletben, a ~ különböző típusú lehet. Ha a bal oldali első bit nem vesz részt a műveletben (az →előjelbit szerepét játssza), →aritmetikai ~ről, ellenkező esetben →logikai ~ről beszélnek. Ha a bitvektor egyik szélén kilépő bitek a bitvektor másik végén felszabaduló bitek helyére lépnek, a műveletet →ciklikus ~nek nevezik.

letapogatási kód (scan code) – IBM-szabványos →billentyűzeteknél használt kód: valahányszor egy billentyűt lenyomnak, ill. felengednek, a billentyűzet áramkörei egy 1

→bájtos –ot állítanak elő, amely egyértelműen azonosítja a jelenséget. A →BIOS megfelelő szubrutinjainak feladata, hogy a –okat a →programok által is használható (→ASCII) információvá alakítsák.

levezető szintaktikai elemzés (derivative syntactical analysis) – →felülről lefelé haladó szintaktikai elemzés

lexikális analízis (lexical analysis) – programok →fordításának v. →értelmezésének első fázisa, amelynek során a →forrásprogramot alkotó karaktersorozatokban az illető forrásnyelv alapelemeit felismerik és a megfelelő szintaktikai osztályba besorolják (→azonosítók, →konstansok, műveleti jelek, →lefoglalt szavak stb.). –sel olyan további információkat is ki lehet szűrni, amelyek megkönnyítik a következő fázis, a →szintaktikai analízis elvégzését. Így pl. a forrásprogram lefoglalt szavai egyenként egyetlen →terminális szimbólumként adhatók tovább a szintaktikai elemzőnek, s így a visszamaradó nyelvtan egyszerűbb és gyorsabb elemzési módszer alkalmazását teszi lehetővé.

lexikális elemző (lexical analyzer, scanner) – →fordítóprogramok, ill. →értelmezőprogramok felépítésében szereplő programegység, amelynek fő feladata a →lexikális analízis elvégzése, de egyéb feladatokat is elláthat: a kódkonverziók, a programsorok szekvencialitásának ellenőrzése, a forrásnyelvű lista előállítása, a programban található megjegyzések kiszűrése stb. Gyakorlatilag lehet önálló programegység, amely egyszeri végrehajtás útján az egész forrásprogramot végigolvassa és a nyelvi alapelemekből a →szintaktikai elemző részére megfelelő formájú közbülső kódot állít elő. Másik gyakorlati megvalósítása a →korutinok elvén alapszik: ilyenkor a szintaktikai elemző hívására mindig a soron következő nyelvi alapelemet szolgáltatja. Függetlenül megvalósítási formájától, a – a nyelvi alapelemeken kívül egy sor olyan információt is közöl a szintaktikai elemzővel, amelyek nélkülözhetetlenek a további elemzésben (→kulcsszó jelleg, →[adat]típusok, kontans értékek, ábrázolási formák stb.).

LIFO-szerkezet (LIFO-structure) – a számítástechnikában igen elterjedt szervezési technika. A struktúrában szereplő elemek olyan rögzítése, amelynek következtében új elemet elhelyezni a szerkezetbe csakis az utolsó elem után lehetséges, onnan kiolvasni v. törölni viszont mindig csak az utolsó elemet lehet. A – jellegzetes megvalósítása a →verem.

lineáris adatstruktúra (linear data structure) – olyan →adatstruktúra, amelynek két kitüntetett adateleme van: az első és az utolsó. Alapvető jellemzője, hogy az adott elemet követő adatelem mindig egyértelműen meghatározható. A – lehet egyirányú (aszimmetrikus), ha az első adatelemtől kiindulva a struktúra bármelyik eleme elérhető, és lehet kétirányú (szimmetrikus), ha bármelyik elem elérhető, függetlenül attól, hogy az első v. az utolsó elemtől indultak ki. Jellegzetes –k pl. a →tömb, a →fa, a →lista stb.

lineáris keresés (linear search) – rendezett adathalmazban történő →keresési algoritmus. Lényege, hogy minden sikertelen összehasonlítás után, a logikailag következő elemet vizsgálják meg, a fizikai tárolási sorrendtől függetlenül. Azonos visszakeresési gyakoriságot feltételezve, a keresési lépések átlagos száma:

$S_n \approx n/2$, ahol n az adathalmazban található összes elem száma.

lineáris nyelvtan (linear grammar) – a →Chomsky-hierarchia szerinti 3 típusú →nyelvtan, amelynél a →helyettesítési szabályok jobb oldalán legfeljebb egy →nemterminális szimbólum szerepelhet, tehát formájuk:

$A \rightarrow \alpha$

$A \rightarrow B\alpha$ (bal lineáris)

$A \rightarrow \alpha B$ (jobb lineáris)

lineáris programozás (linear programming) – minden olyan feladat megoldása, amelyben egy lineáris függvény maximumát (v. minimumát) kell meghatározni valamilyen lineáris egyenlőtlenség-rendszer nemnegatív megoldásai mellett. Jellegzetes –i feladat annak az n számú tevékenységnek mennyiségi meghatározása egy gazdasági rendszerben, ahol adott számú erőforrás adott korlátok között a rendszer nyereségét maximálja, ha a tevékenységek méretei folytonosak, az erőforrások normái és a fajlagos nyereségek állandók, a felhasznált erőforrások és az elért nye-

reség egyenesen arányosak a tevékenység méreteivel.

LIPS (Logical Inferences Per Second) – →logikai programozás útján létrehozott programtermékek (pl. →szakértőrendszerek) műveleti sebességének mértékegysége (logikai következtetés másodpercenként).

LISP (LISt Processing) – listák kezelésére kidolgozott magas szintű →programozási nyelv, amely egyaránt alkalmas nemnumerikus, vegyes numerikus és szimbolikus adatok feldolgozására. Problématerületére jellemző az olyan adatok használata, amelyeknek a mérete és értéke a feladatok megoldásához közeledve dinamikusan változik. Az ilyen típusú adatok ábrázolására legalkalmasabbak a listastruktúrák, amelyekkel rendszerint iteratív és rekurzív algoritmusok végezhetők. Speciális felhasználási területek a matematikai bizonyítások gépi ellenőrzése, induktív következtetések, alakillesztő lánctranszformációk, áramkörelmélet, játékelmélet, szimuláció, nyelvészet stb. A ~ben a programok és az adatok szimbolikus kifejezések formájában vannak megadva (→S-kifejezések). Az elérhető memóriamennyiség nagy részét S-kifejezések listaszerű tárolására használják. A ~ csak kétfajta adatot ismer: →atomot és →listát. Az atomok a feldolgozás szempontjából oszthatatlan entitások, amelyekhez különböző attribútumok kapcsolhatók, szövegszerűen →numerikus és →nemnumerikus literálok.

lista (list) – az egyik legfontosabb, véges számú elemmel rendelkező →lineáris adatstruktúra. Legfontosabb tulajdonságai: létezik egy olyan összefüggés az adatelemek között, amely minden adatelemre vonatkozóan meghatározza az azt megelőző, ill. követő adatelemet; létezik egy olyan adatelem, amely megelőz minden más elemet a struktúrában; létezik egy olyan adatelem, amelyet nem követ egyetlen más elem sem a struktúrában.

Az →operatív memóriában v. a →háttértárban a ~ ábrázolására legalkalmasabb a →láncolás, mert így az adatelemek fizikai elhelyezése függetleníthető az adatstruktúra logikai felépítésétől. Az egyszerű ~ elemei →atomok. A hierarchikusan strukturált ~ elemei viszont lehetnek atomok és allisták egyaránt. Ez utóbbi igen alkalmas a →lengyel jelölésű aritmetikai kifejezések ábrázolására.

listafeldolgozás (list processing) – →lista típusú →lineáris adatstruktúrákkal dolgozó feladatok számítógépes megoldása. Felhasználási területei elsősorban a matematikai logika, a mesterséges intelligencia, a szöveges információk feldolgozása, a nyelvészet v. általában minden olyan terület, ahol a feldolgozandó adatok ábrázolására előnyösen felhasználható a listastruktúra. A ~ végrehajtható közvetlen módon, vagyis a láncolt ábrázolású struktúra →mutatóival végzett explicit műveletekkel v. pedig szimbolikus módon, specializált programozasi nyelv segítségével, amely megfelelő utasításkészlettel rendelkezik a listákkal végzett műveletek végrehajtására (→LISP).

listázás (listing) – információ szöveges megjelenítése.

literál (literal) – 1. Egyes →programozási nyelvekben a →konstanssal rokon értelemben használt elnevezés olyan programozási objektumokra vonatkozóan, amelyeknek változatlan értékét szövegszerű ábrázolásuk (a felépítő karakterek értéke és sorrendje) határozza meg a program végrehajtásának egész idejére. A ~ lehet numerikus és alfanumerikus, a felépítő karakterek jellegének megfelelően. – 2. Egyes programozási nyelvek alapeleme, amely →jelsorozatok szerepeltetését teszi lehetővé kifejezésekben, argumentumokban, utasításokban. Leggyakrabban egyszeres v. kétszeres idézőjel közé kell zárni. A →COBOL programozási nyelvben pl. alapvető különbség van a következő két objektum között:

1234.56 és „1234.56"

logaritmikus keresés (logarithmic search) – →bináris keresés

logikai [adat]állomány (logical file) – olyan →[adat]állomány, amelynek előállítására és kezelésére az →operációs rendszer speciális adatkezelő szubrutinokat tartalmaz. Az operációs rendszer (akárcsak a felhasználó) a ~t →logikai rekordok halmazaként tekinti, amely előre meghatározott szabályok szerint van rendezve és tárolva az →adathordozón. Legfontosabb típusai: a →szekvenciális [adat]állomány, az →indexszekvenciális [adat]állomány és a →direkt szervezésű [adat]állomány, ill. a →particionált [adat]állo-

mány. A felhasználó szempontjából a ~hoz való hozzáférési igény egysége a logikai rekord, a →blokkok tördelése, ill. a rekordoknak blokkokba tömörítése az operációs rendszer feladata (→blokkolás).

logikai adatszerkezet (logical data structure) – az adatfeldolgozásban szereplő információknak és a közöttük fennálló kapcsolatoknak a felhasználó (programozó) felfogásában történő visszatükröződése, függetlenül azoknak fizikai ábrázolásmódjától, →adathordozókon való rögzítési módjától. A ~ alapja az a felismerés, hogy az adatok a számítógépes ábrázolástól függetlenül is rendelkeznek leírható és dinamikusan tanulmányozható tulajdonságokkal. A magasabb logikai szintű leírás lehetővé teszi a megfelelő szintű logikai kezelést is. A ~ és fizikai megvalósítása közötti függetlenség áll pl. az →adatbázisokon alapuló adatfeldolgozás központjában (→logikai-fizikai leírás).

logikai [adat]típus (logical [data] type) – →Boole-[adat]típus

logikai alapú módszer (logic-based method) – →ismeretábrázolásban használható programozási módszer, amely a predikátumkalkulust alkalmazza az →ismeretek strukturálására és a végrehajtás irányítására (→logikai programozás, →Prolog).

logikai algebra (logical algebra) – →Boole-algebra

logikai cím (logical address) – A →forrásnyelvű programban szereplő →lokális szimbólumok és →globális szimbólumok kettős szerepet játszanak: egyrészt az illető programozási objektumok →azonosítójaként lehetővé teszik azok felhasználását a feldolgozást megtestesítő utasításokban, másrészt az objektumok ~ét alkotják. A programot csak akkor lehet ténylegesen végrehajtani, ha ezeket leképezik az →operatív memória fizikai címeire. A leképzés általában több lépésben valósul meg, a →fordítás, a →kapcsolás, a →betöltés, esetleg a végrehajtás közben (pl. a →virtuális címzésnél).

logikai címtartomány (logical address space) – az az absztrakt →memória, amelyben →program írásakor a programozó gondolkozik, programjának →címeit kialakítja, elhelyezi. A kész programot ezután – gyakran jelentős további ráfordítással – az adott konkrét számítógép véges lehetőségeihez, az →operatív memória méretéhez kell igazítani (→címtartomány, →virtuális memória).

logikai egység (logical unit) – programozástechnikai értelmezésben: a →moduláris programozás módszertanában használt fogalom, olyan programegység neve, amely jól meghatározott logikai funkciót hajt végre. Az ilyen programegység (modul) könnyen érthető, logikája tiszta, világos, áttekinthető. A jó minőségű ~ a programozási osztály szabványai szerint kódolt, jól tesztelhető modul, ez biztosítja, hogy az ilyen modulokból felépülő programtermék is megfelelő minőségű lesz.

logikai érték (logical value) – a →logikai változó v. →logikai kifejezés felvett értéke; mint a logikai algebrában, csak az „igaz" v. a „hamis" értékek egyike lehet.

logikai eszköznév (logical device name) – →periferikus műhöz kapcsolt szimbolikus név, amelyet a programozó úgy használhat fel programjában, hogy nem kell feltétlenül tudnia, melyik fizikai eszközre vonatkozik ténylegesen. A ~ leképzését tényleges fizikai eszközcímmé az →operációs rendszer végzi el.

logikai-fizikai leírás (logical-physical description) – adatoknak és a közöttük fennálló kapcsolatoknak különböző értelmezési szintek szerinti leírása, tekintettel arra, hogy az adatok szervezése (és értelmezése) eltérő lehet a programozó adatfelfogásában és a fizikai megvalósításban. A logikai jelző ilyen értelmezésben a felhasználó látásmódját, a fizikai pedig az →adathordozókon való rögzítés módját érzékelteti. Egyszerű példa e különbség illusztrálására az →[adat]állományok logikai értelmezése és fizikai megvalósítása. A →mágneslemezen tárolt fizikai bejegyzések több logikai rekordból épül(het)nek fel (→blokkolás), s egymással láncolva vannak. A programozónak a logikai rekordokra a láncolás sorrendjében van szüksége, de a láncolás fizikai megoldása nem érdekli. Az →[adat]állomány-kezelő rendszer, ill. az →adatbázis-kezelő rendszer feladata az, hogy a programozó logikai szintű utasításait és adatleírásait átalakítsa a fizikai valóság értelmezésére. Az adatleírás logikai, ill. fizikai szintjei közötti elhatárolás elsősorban az →adatbázisok szervezésében kapott különleges

jelentőséget, mert csak így érhető el a magas szintű →adatfüggetlenség.

logikai függvény (logical function) – 1. Kétértékű (igaz v. hamis) logikai értékkészleten definiált függvény. Bármelyik ~ megadható v. a megfelelő →logikai kifejezéssel v. az argumentumok összes lehetséges kombinációját és a hozzá tartozó függvényértéket tartalmazó →igazságtáblázattal. – 2. Programozástechnikai értelmezésben: olyan →függvény (mint programegység), amelynek csak logikai értéke lehet. Logikai kifejezésekben operandusként szerepelhet, aktivizálásához a szükséges gépi műveleteket a →fordítóprogram, ill. az →értelmezőprogram generálja.

logikai hiba (logical error) – →programozási hibák egyik fontos típusa. →Fordítóprogram (ill. →értelmezőprogram) általi felfedezése általában nem biztosítható, csak rendszeres tesztelés útján szűrhető ki a hibásan működő programból. Leggyakrabban előforduló ~: a logikai döntés ágainak felcserélése; a nem teljes logikai döntés; a hibás ciklusszervezés; az egyes határesetek hibás kezelése v. figyelmen kívül hagyása stb.

logikai hozzáférés (logical access) – →adatbázisokban tárolt adatok logikai szintű elérési módszere, amelynek segítségével a felhasználói program és az adatbázis közötti kapcsolat megvalósul. A ~ jellegzetes műveleteinek leírására az →adatkezelő nyelvet használják. A felhasználói program által kifejezett igényeket az →adatbázis-kezelő rendszer elégíti ki az →adatbázisséma, az →adatbázis-alséma, valamint a →fizikai adatbázis leírása alapján. A művelet során a felhasználó által kért logikai rekord teljesen transzparens módon kerül a program munkaterületére, ahol a felhasználói program már kezelni tudja. Hasonló módon, ha egy logikai rekordot be kell vinni az adatbázisba, a felhasználói program elhelyezi azt a munkaterületre, és adatkezelő nyelven kiadja a megfelelő utasítást. A tényleges fizikai hozzáférés az adatbázis-kezelő rendszer és az →operációs rendszer feladata.

logikai kifejezés (logical expression) – magas szintű →programozási nyelvekben logikai operandusokból és műveleti jelekből felépített olyan sorozat, amely logikai értéket előállító kiértékelést jelent. Az operandusok →logikai [adat]típusú konstansok, változók és függvényeljárások v. →relációs kifejezések. A kiértékelésben elsőbbségi (precedencia-) szabályok érvényesek; az elsőbbségi sorrend csökkenő irányban: relációs kifejezések, →negációk, →logikai szorzások, →logikai összeadások. Az aritmetikai kifejezésekhez hasonlóan a műveleti sorrendet itt is befolyásolni lehet megfelelő zárójelezéssel.

logikai konstans (logical constant) – olyan →konstans, amely csak az „igaz" v. a „hamis" értékekkel rendelkezhet, s mint ilyen, logikai, ill. döntési műveletekben vehet részt.

logikai léptetés (logical shift) – a →léptetés műveletének egyik változata, amely a bitvektor minden bitjét azonos módon kezeli (tehát nem veszi figyelembe az →előjelbitet, mint az →aritmetikai léptetés). Ciklikus ~nél a bitvektor egyik oldalán kilépő bitek a másik oldalon ugyanolyan sorrendben megjelennek. Nem ciklikus ~nél a bitvektor egyik oldalán kilépő bitpozíciók tartalma elvész, a másik oldalon felszabaduló bitpozíciók pedig nullával töltődnek fel.

logikai összeadás (logical addition) – (v. diszjunkció) kétargumentumú →logikai függvény, amelynek értéke akkor igaz, ha legalább az egyik argumentum értéke igaz. Igazságtáblázata:

v	0	1
0	0	1
1	1	1

logikai programozás (logical programming) – a →mesterséges intelligencia területén végzett kutatások, a következtető rendszerek megvalósítására tett törekvések eredményeként megjelent új programozástechnikai irányzat. Ennek egyik ága a predikátumkalkulus-programozás, másik ága a funkcionális programozás; mindkettőnél maguk a logikai formulák alkotják a programozási nyelvet. Az új irányzat központi célkitűzése az, hogy a feladatot logikai állítások formájában lehessen megfogalmazni, a megoldást pedig tételbizonyító program állíthassa elő. Az eddigi kutatások szerint a matematikai logikát nem kell teljes általánosságában felhasználni, hanem ki lehet választani egy egyszerűbb résznyelvet, amelyhez gyors, hatékony működésű tételbizonyító eljárások készíthetők. Egy ilyen rész-

nyelvet R. Kowalsky és P. Hayes dolgozott ki a Horn-formulák alapján, majd a gyakorlatban A. Colmerauer valósította meg (→Prolog).

A Horn-formulák használata a logikának olyan korlátozása, amelyben csak a „B akkor igaz, ha $A_1, A_2, \ldots, A_n$ mind igazak" alakú formulák vannak megengedve. Egy ilyen formulának algoritmikus értelmezés is adható: „B végrehajtása $A_1, A_2, \ldots, A_n$ egymás utáni végrehajtásából áll".

logikai rekord (logical record) – a felhasználó szemléletében az →[adat]állomány legkisebb logikai egysége, amelynek meghatározott belső szerkezete és a hivatkozást egyértelművé tevő azonosítója van. Rendszerint nem egyezik meg teljesen a →fizikai bejegyzéssel, ami tkp. a ~ fizikai ábrázolása valamely →adathordozón. Ez az állítás elsősorban a mágneses adathordozókra vonatkozik, ahol több ~ a fizikai szervezést elősegítő járulékos adattal együtt alkotja a fizikai bejegyzést (→blokkolás).

logikai szegmens (logical segment) – →lapkezeléssel megvalósított →multiprogramozásnál a felhasználói program logikai egysége (megkülönböztetésül az →átfedési szerkezettel kapcsolatban használt →szegmens fogalmától). Rendszerint speciális →kapcsolóprogram állítja össze, amely minden ~nek nevet ad (rendszerint a ~ programon belüli sorszáma). A program hivatkozásai kétkomponensű logikai címek, ahol az első komponens a ~ neve, a második pedig a hivatkozás szegmensen belüli relatív címe (ez szintén két részből épül fel, a →lap sorszámából és a lapon belüli relatív címből). A ~nek ez a formája bárhol elhelyezhető az →operatív memóriában. A logikai címek leképzése fizikai címekre dinamikusan valósul meg, a szegmenstáblázatok, ill. laptáblák segítségével (→virtuális címzés).

logikai szorzás (logical multiplication) – (v. konjunkció) kétargumentumú →logikai függvény, amelynek értéke akkor és csakis akkor igaz, ha mindkét argumentum értéke egyszerre igaz. Igazságtáblázata:

∧	0	1
0	0	0
1	0	1

logikai változó (logical variable) – olyan →változó, amely csak az „igaz" v. a „hamis" értékeket veheti fel, s mint ilyen, logikai és döntési műveletekben vehet részt.

LOGO - oktatási célokra kifejlesztett egyszerű, de meglepően hatékony →programozási nyelv. Alapvető jelentősége, hogy szinte rákényszeríti a tanuló gyermeket az algoritmikus gondolkozásra. Technikatörténeti jelentősége, hogy bevezette a →teknőcgrafika néven ismertté vált módszert, amelynek révén a tanulók játékos módon alapos grafikai ismereteket sajátíthatnak el.

lokális szimbólum (local symbol) – blokkszerkezetű (→PL/1, →ALGOL, →Pascal stb.), ill. modulszerkezetű (→asszembler nyelvű) programokban használt →azonosító, amelynek érvényességi tartománya az a blokk (ill. modul), ahol a →deklarációja is található. A ~ra csakis a deklarációt is tartalmazó programegységben lehet hivatkozni (szemben a →globális szimbólummal).

LY

lyukkártya (perforated/punched card) – információ tárolására szolgáló, papírból készült, szabványos méretű kártya, amelyen az információt a téglalap (egyes rendszerekben kör) alakú perforációk hordozzák. A leggyakrabban használt ~n 80 oszlopon egyenként 12 lyukasztási hely van, s rendszerint egy-egy oszlop egyetlen alfanumerikus karaktert hordoz (a számjegyeket egy, a betűket kettő, a speciális karaktereket kettő v. három lyukasztás rögzíti). A →kötegelt feldolgozású számítógéprendszerek egyik legfontosabb beviteli adathordozója volt.

lyukszalag (perforated/punched tape) – információ tárolására szolgáló, többnyire papírból v. műanyagból készült szalag, amelyen az információt perforációk hordozzák. Egy-egy alfanumerikus karaktert (→ASCII- v. →EBCDIC-kódolásban) a ~ mozgására merőlegesen perforált lyukkombináció hordozza.

M

M – számítástechnikai vonatkozásban mértékegységek (pl.tárolási kapacitás) kifejezésére használt egyezményes jel, a mega rövidítése. Értéke: 1 **M**egység = 1024 * **K**egység = 1024 x1024 = 1 048 576 egység.

magas szintű programozási nyelv (high level programming language) – olyan →programozási nyelv, amely megfelelő eszközökkel rendelkezik adatstruktúráknak, feldolgozási és vezérlési szerkezeteknek konkrét számítógéptől független leírására. A ~eken kódolt programok csak →fordítás v. →értelmezés útján hajthatók végre a számítógépen. A ~ek jelölésmódja közel áll az adott terület klasszikus jelölésrendszeréhez, így a programozó könnyen, viszonylag gyorsan meg tudja oldani feladatát. Az első jelentősebb ~ek az ötvenes években alakultak ki (→FORTRAN, →ALGOL, →COBOL stb.), számuk azóta jóval meghaladja a természetes nyelvekét. Elterjedésük elsősorban annak köszönhető, hogy a nyelvészeti és programozástechnikai kutatások eredményeképpen ma már igen hatékony, sokszor automatizált eszközök állnak rendelkezésre újabb és újabb →fordítóprogramok és →értelmezőprogramok előállítására.

mágneskártya (magnetic card) – mágnesezhető anyaggal bevont, kártya alakú →adathordozó.

mágneslemez (magnetic disk) – 1. Mágnesezhető anyaggal bevont, forgó acéllemezekből álló →adathordozó. Minden felülethez legalább egy író-olvasó fej tartozik. Az adatok tárolása a különböző felületeken koncentrikus kör alakú pályákon, →sávokon történik. A különböző felületeken egymás felett elhelyezkedő és így az író-olvasó fej elmozdítása nélkül egyidejűleg elérhető sávok együtt →cilindert (palástot) alkotnak. Fixfejes kivitelezésnél minden sávhoz egy-egy külön író-olvasó fej tartozik, mozgófejes formánál minden lemez fölött sugárirányú mozgással egyetlen író-olvasó fej az összes sávot ellátja. – 2. Modernebb formája a →merevlemez és a →hajlékonylemez.

mágneslemezes memória (magnetic disk storage) – →adathordozóként →mágneslemezt használó, közvetlen hozzáférésű →háttértár.

mágnesszalag (magnetic tape) – információ tárolására alkalmas, mágneses anyaggal (rendszerint vas-oxiddal) bevont műanyag szalag, amelyen az információt a mágneses bevonat állapota hordozza.

mágnesszalagos operációs rendszer (tape operating system) – →operációs rendszerek olyan típusa, amelynél a nemrezidens részt →mágnesszalagon tárolják, és az ott rögzített rendszerprogramok végrehajtásához is mágnesszalagra van szükség.

mágnesszalagos szervezés (magnetic tape organization) – információk →mágnesszalagon történő tárolását meghatározó szabályok összessége. A mágnesszalag elejét és végét egy-egy szalagkezdő, ill. szalagvégjel jelöli, közöttük helyezkednek el a programok által rögzíthető, leolvasható és megváltoztatható információk. Ha a mágnesszalag →[adat]állományként van megszervezve, az adathordozó, ill. a rajta tárolt [adat]állomány(ok) azonosítására →szabványos címkék szolgálnak (→adathordozói címke, →[adat]-állomány-kezdőcímke, →[adat]állomány-zárócímke, felhasználói címkék stb.). Egy mágnesszalagtekercs (→kötet) általában egy [adat]állományt tartalmaz, de esetenként rögzíthető rajta több [adat]állomány is, ill. egy v. több [adat]állomány elfoglalhat több tekercset is. A ténylegesen végrehajtott →b/k műveletek során →blokkokat írnak v. olvasnak, ez az információcsere egysége. A blokk egy v. több →logikai rekordot tartalmazhat (→blokkolás), a blokkokat →blokk-közök választják el egymástól. Szabványos címkék használatánál ezek értelmezése és kezelése az →operációs rendszer (→[adat]állomány-kezelő rendszer) feladata. A nemszabványos, ill. felhasználói címkék kezeléséről a felhasználónak kell gondoskodnia.

A legtöbb magas szintű programozási nyelvben megtalálhatók az alapvető mágnesszalag-kezelő utasítások: READ (a következő blokk olvasása), WRITE (az új blokk kiírása), esetleg BACKSPACE (visszalépés egy blokkal) stb.

magyarázó modell (explanatory model) – olyan →modell, amely a jelenségek leírása során kimutatja az okozati összefüggéseket is (szemben a →leíró modellel).

magyarázó szöveg (comment) – programok →forrásnyelvű szövegében megjelenhe-

tő, a program dokumentálását, magyarázatát szolgáló karaktersorozat. A →fordítás során teljesen eltűnik a programból, a végrehajtás szempontjából semmi jelentőséggel nem rendelkezik (a →tárgyprogramban semmi nyoma nem marad).

makroasszembler (macroassembler) – →makrofeldolgozási lehetőségekkel rendelkező →asszembler, amely lehetővé teszi a programozó számára, hogy →makrodefiníciókkal új utasításokat, →makroutasításokat definiáljon, s ezáltal az illető nyelvet igényeinek megfelelően kibővítse.

makrodefiníció (macrodefinition) – olyan →deklaratív utasítás, amelynek segítségével nevet rendelnek egy meghatározott utasítássorozathoz. Rendszerint speciális →pszeudoutasítással kezdődik, ez jelzi, hogy a következő sor a →makroutasítás neve lesz. A makroutasítás nevét az esetleges →formális paraméterek kísérik. Ezután következnek azok az utasítások, amelyek a majdan behelyettesítendő szöveget, a makroutasítás törzsét alkotják. A ~t a makrotörzs végét jelző újabb pszeudoutasítás zárja. A ~ban rögzített név a ~ törzsében leírt utasítássorozatot képviseli a →fordítás egész menete alatt.

makrofeldolgozás (macroprocessing) – legáltalánosabb értelmezésben valamely →programozási nyelv →szintaktikai kiterjesztése a felhasználói igények alapján. Konkrétan jól meghatározott utasítássorozat egyetlen új utasításos helyettesítése. A →makroutasítás alkalmazásához a programozónak előzetesen definiálnia kell azt a műveletsort, amelyet új utasításnévvel akar ellátni (→makrodefiníció). Ettől kezdve ez a név az egész utasítássorozatot képviseli, a →fordítás során valahányszor előfordul a forrásprogramban (→makrohívás), a →fordítóprogram behelyettesíti a megfelelő utasításokat.

Előre rögzített makrodefiníciók révén a legtöbb →asszembler rendelkezik bizonyos ~i lehetőségekkel. Ezeknél a makrodefiníciókat speciális →makrokönyvtár tartalmazza, a makroutasítások nevei pedig →kulcsszó jellegűek az asszembler számára. A →makroasszemblerek felhasználói makrodefiniálást is lehetővé tesznek.

Az →operációs rendszereknek konkrét →konfigurációra történő adaptálását rendszerint ~ teszi lehetővé (→rendszergenerálás). Ugyancsak fontos szerepe lehet a ~nak egyes programtermékek →hordozhatóságának biztosításában. A ~t magas szintű programozási nyelveknél is lehet használni. (Lényegében erre az elvre alapul az →előfordítás bevezetése is.)

makrohívás (macrocall) – a →makrodefinícióban szereplő név (esetleg →aktuális paraméterekkel kiegészített) szövegszerű megjelenése a →forrásprogramban, ennek hatására a →fordítóprogram a makrodefiníció törzsében meghatározott utasítássorozatot a forrásprogramba illeszti.

makrokifejtés (macroexpansion) – az a tevékenység, amelynek során a →fordítóprogram (→asszembler) a forrásprogramban megjelenő →makrohívást feldolgozza, vagyis a makrohívás helyére a →makrodefinícióban szereplő utasításcsoportot beilleszti. Az esetleges →aktuális paraméterek lehetővé teszik a bemásolandó szöveg módosíthatóságát (természetesen ez csak a makrodefinícióban szereplő →formális paraméterek útján lehetséges).

makrokönyvtár (macrolibrary) – →makrodefiníciókat tartalmazó speciális →könyvtár.

makromeghatározás (macrospecification) – adatstruktúráknak, utasítássorozatoknak v. általában szövegeknek már ismert, egyszerűbb elemek segítségével történő meghatározása →makrodefiníciók alapján.

makronyelv (macrolanguage) – olyan →programozási nyelv, amely →makroutasítások definiálására alkalmas eszközökkel rendelkezik. Jelentősége, hogy a meglévő nyelvet a programozó saját igényeinek megfelelően módosíthatja, bővítheti új utasítások definiálásával. Az →asszembler szintű ~ segítségével a különböző feladatok tömören megfogalmazhatók és aránylag könnyen programozhatók (akárcsak a magas szintű nyelveknél), ugyanakkor teljes egészében kihasználhatók mindazok az előnyök is, amelyeket az asszembler nyelv használata jelent.

makroparaméter (macroparameter) – a →makroutasítások mint →nyílt szubrutinok leírására szolgáló →formális paraméter v. a makroutasítások konkrét aktivizálására (hívására) használt →aktuális paraméter.

makroprocesszor (macroprocessor) – olyan program, amely lehetővé teszi szöve-

gek generálását egy forrásszöveg alapján, a →makrofeldolgozás módszerének alkalmazásával. A legismertebb — a →makroasszembler. Szövegszerkesztésekre, →fordítóprogramok gyakorlati megvalósítására, →gépi fordításra, programtermékek generálására stb. használható. Elvileg az →absztrakt számítógép és egy konkrét rendszer közötti kapcsolat megteremtője.

makroutasítás (macroinstruction) – →nyílt szubrutin (utasításcsoport) egyetlen soros rövidítése, amelyet a →fordítóprogram mindannyiszor beépít a forrásprogramba, valahányszor →makrohívást talál a lefordítandó eredeti szövegben. Általában (de nem kizárólagosan) az →asszembler nyelvekre jellemző. A ~hoz tartozó →makrodefiníció lehet implicit, amikor a felhasználó a fordítóprogram által definiált ~okat használhatja az előírt formában (pl. →rendszer-makroutasítás), v. lehet explicit, amikor a programozó maga definiálhatja saját ~ait, s ezáltal megfelelő, magasabb szintű programozási lehetőségeket teremthet igen egyszerű eszközökkel.

mantissza (mantissa, floating-point coefficient) – →lebegőpontos számábrázolásban használt bináris vektor, a számok

$$\pm m \cdot a^{\pm e}$$

formában történő felírásánál az m komponens neve.

második generációs számítógép (second generation computer) – kb. 1955 és 1965 között elektroncsöves és diszkrét félvezetős elektronikával (tranzisztorokkal) épített számítógép (kapcsolási sebessége mikrosec nagyságrendű). A tranzisztorizált →architektúra előnye a nagyobb megbízhatóság, a kis méret, a nagy műveleti sebesség. Alapvető jellemzője a nagy sebességű →ferritgyűrűs →operatív memória. Szervezési szempontból legfontosabb újítása a →b/k műveletek aszinkronizálása, az →adatcsatorna megjelenése. Ugyancsak ennél a generációnál jelent meg a →megszakítási rendszer. →Szoftvervonatkozásban az első tényleges →operációs rendszer megvalósításában hozott újat, s ugyancsak ebben a periódusban jelentek meg az első →programozási nyelvek.

másodlagos csatorna (secondary channel) – az adatátvitelt ténylegesen megvalósító →kommunikációs csatornához rendelt, de fizikailag teljesen független csatorna, amelynek feladata az adatátvitelt vezérlő információk átvitele.

maszk (mask) – megfelelő hosszúságú bitvektor, amely a →maszkolás műveletével egy adat megfelelő bináris pozícióit kiemeli az adatból.

maszkolás (masking) – →bit szintű adatfeldolgozásra jellemző művelet. Tkp. az érintett adat és egy →maszk (megfelelő tartalmú bitvektor) →logikai szorzata, amelynek révén a maszkban található 1-es bitpozíciók az adat megfelelő bitpozícióin található értékeket kiemelik.

maszkolt megszakítás (masked interruption) – a →megszakítási rendszerben egyébként jelenlévő, de pillanatnyilag kiiktatott állapotban található →megszakítás. Az erre vonatkozó igények feldolgozása későbbi időpontra halasztódik. Ha ugyanis egy előző megszakítást feldolgozó szubrutin végrehajtása közben fellépne egy újabb, ugyanahhoz a →megszakítási szinthez tartozó megszakítás, elveszne az előző →programállapotszó, és az illető programhoz többé nem lehetséges semmilyen módon a visszatérés.

matematikai logika (mathematical logic) – a logikus gondolkodás törvényeit matematikai módszerekkel tanulmányozó tudományág. Programozáselméleti szempontból különösen jelentős az új elvű, igen magas szintű nyelvek kifejlesztésében (→Prolog, →logikai programozás).

megbízhatóság (reliability) – a számítógéprendszernek (mint →hardver és →szoftver →erőforrások együttesének) az a tulajdonsága, hogy bizonyos ideig hibamentesen működik. Igen lényeges mutató, ugyanis esetleges hardver-, ill. szoftverhibák egész felhasználói közösség munkáját tehetik bizonytalanná v. lehetetlenné. Ahol a ~ – az alkalmazás természetéből adódóan – különös hangsúllyal esik latba, a →redundancia növelésével érhetik el a megfelelő ~i szintet. Programozástechnikai szempontból a ~ magában foglalja a következő követelményeket is: a program a javítható bemeneti hibák javítását automatikusan, figyelmeztetéssel egybekötve végezze el; az előforduló →túlcsordulásokat tartsa számon, jelezze és esetleg javítsa

ki; a program a kitűzött feladatot ésszerű időkorláton belül hajtsa végre, ill. az esetleg felmerült akadályokról idejében és kimerítően értesítse a felhasználót.

megelőző karbantartás (preventive maintenance) – számítógéprendszerek →megbízhatóságát növelő műveletek összessége. Előre megállapított időszakonként az egész rendszert leállítják, és speciális →diagnosztizáló programok segítségével kiszűrik azokat a funkcionális egységeket, amelyek paramétereik határértékén működnek.

megengedett hiba (tolerable error) – az a maximális →hiba, amely mellett a kapott eredmény még gyakorlatilag elfogadható.

meghajtó (driver) – →operációs rendszerekhez tartozó →szubrutin, amely a számítógéphez csatlakozó egyes →periferikus műveket vezérli. Ez a vezérlés alapjában véve fizikai szinten valósul meg, az illető eszköz eredeti működésének figyelembevételével. Bizonyos értelemben →interfészként is felfogható, ugyanis a ~ szabványosított →protokollok alapján valósítja meg az adatátvitelt a rendszer és az illető eszköz között.

megjelenítő (displaying device) – információk közvetlenül értelmezhető ábrázolására szolgáló, a számítógéppel közvetlen kapcsolatban álló berendezés. Az információk ábrázolása történhet csak alfanumerikus karakterekkel (alfanumerikus ~) v. grafikus és alfanumerikus karakterekkel (grafikus ~), általában katódsugárcsöves képernyőn. Egyszerűbb berendezéseknél (pl. zsebszámológépeknél) használhatnak fénydiódákból v. folyadékkristályokból összeállított (7-16) szegmenst is az alfanumerikus karakterek rajzolatának kialakítására. Katódsugárcsöves ~nél a karakterábrázolásra a következő alapvető eljárásokat szokták használni: a katódsugárcső minden előállítható karakter számára megfelelő sablont tartalmaz; a karakterek megfelelő utasítások szerint pontokból állnak elő (nyomtatási raszter); a karaktereket olyan vonalkákból állítják elő, amelyeknek hossza és szöge változó; a karakterek rajzolata vonalakból és ívekből áll össze.

megőrzési periódus (retention period) – az a – rendszerint napokban v. hónapokban mért – időperiódus, amelynek tartamára a mágneses →adathordozón rögzített információk (adatok) törlését megtiltják. →[Adat]állományokra vonatkoztatva a ~ csak az [adat]állománynak mint logikai egységnek a megsemmisítését zárja ki, az aktualizálási v. karbantartási műveleteket el lehet végezni. Szabványos szervezésű [adat]állományoknál a ~t az →[adat]állomány-kezdőcímkében rögzítik. A ~on belül az [adat]állomány törlését csak a felhatalmazott felhasználó kifejezett kívánságára lehet elvégezni, az egész művelet pontos ellenőrzése az →[adat]állomány-kezelő rendszer egyik fontos feladata.

megszakítás (interrupt) – a →hardver és a →szoftver határterületét érintő fogalom. Valamely külső v. belső esemény következménye, s a →felügyelőprogram vezérlésére szolgál. Vezérlő szerepe abban nyilvánul meg, hogy hatására az éppen végrehajtás alatt lévő folyamat megszakad, a rendszer felügyelő állapotba kerül, a vezérlés pedig a ~t elemző programrésznek adódik át. A ~ kezelése olyan módon zajlik le, hogy a felfüggesztett folyamat később folytatható legyen.

Legfontosabb típusai közül a gépi ~okat hardverhibák (pl. feszültségkiesés, paritáshiba, adatcsatorna hibája stb.) okozzák; a programhibából eredő ~ oka →csapda (ismeretlen utasításkód, hibás v. védett memóriacímre való hivatkozás, aritmetikai →túlcsordulás, nullával való osztás stb.) lehet; programfeltétel-~ akkor keletkezik, ha maga a végrehajtás alatt álló program kér ~t, mert szüksége van az →operációs rendszer szolgáltatásaira (→rendszer-makroutasítás, →felügyelőprogram-hívó utasítás); a →periferikus művek ~ait az adatforgalommal, a berendezések állapotával stb. kapcsolatos állapotváltozások eredményezik.

megszakításfeldolgozó szubrutin (interrupt service subroutine) – →operációs rendszerekhez tartozó olyan →szubrutin, amelynek aktivizálása automatikusan megy végbe a rendszerben létrejövő valamilyen →megszakítás következtében. Feladata a megszakítás típusának meghatározása és az annak megfelelő műveletek elvégzése. A megszakítás feldolgozásának fontosabb műveletei: az éppen aktív program állapotának kimentése későbbi folytatás céljára (→programállapotszó, regisztertartalmak kimentése); a megszakítási szint maszkolása (→maszkolt meg-

szakítás) a feldolgozás idejére; a megszakítás kódjának kiértékelése, a megfelelő tevékenységek aktivizálása; a felfüggesztett programok közül a legmagasabb →prioritással rendelkező állapotának visszaállítása és újraindítása.

megszakítási cím (interrupt address) – az az →operatív memóriabeli cím, amelyre a vezérlés automatikusan áttevődik, amikor a számítógéprendszerben →megszakítás jön létre. Rendszerint v. közvetlenül a megszakításnak megfelelő szubrutin kezdőcíme v. az eseményt elemző és megfelelően intézkedő szubrutin címe. Az első típusra jellemző pl. a →lebegőpontos műveletek programozott megoldása, amikor hiányzik a megfelelő aritmetikai egység, a másodikra pedig a →felügyelőprogram-hívó utasítások kiértékelése és végrehajtása.

megszakítási rendszer (interrupt system) – valamely számítógéprendszer vezérlését megvalósító alrendszer, amely a rendszerben létrejövő véletlenszerű állapotváltozásokra hívja fel a →felügyelőprogram figyelmét. Az állapotvizsgálat elvileg úgy is elképzelhető, hogy a rendszer periodikusan megvizsgálja önmagát, de ez a módszer komplexebb rendszereknél egyre időigényesebb és összetettebb folyamat lenne. Éppen ezért a gyakorlatban a rendszer csak akkor végzi el a vezérlés folyamatának megváltoztatásához szükséges állapotvizsgálatot, amikor az állapotváltozás már megtörtént. Természetesen a különböző típusú →megszakítások nem azonos fontosságúak a szükséges beavatkozás tekintetében, ezért az egyes megszakításokat elsőbbségi (prioritási) csoportokba sorolják (ezek pedig további prioritási sorrendbe rendezhető v. várakozásra ítélhető egyenrangú eseményekre tagolhatók). Azért, hogy egy megszakítás ne eredményezhesse hasznos információk elvesztését, az éppen végrehajtás alatt lévő folyamat csak bizonyos, ún. megszakítható pontokban függeszthető fel (pl. két utasítás között, →megszakítható utasítás megfelelő pontjaiban stb.), és csakis akkor, ha egy magasabb prioritású esemény kezdeményezte az újabb megszakítást.

megszakítási szint (interrupt level) – a →megszakítási rendszerekkel kapcsolatos fogalom. A →megszakításokat sürgősségük (fontosságuk) alapján csoportokba, ~ekbe sorolják, és prioritásokat rendelnek hozzájuk. Többszintű megszakítási rendszerben a kéréseket a hozzájuk rendelt prioritások sorrendjében veszik figyelembe. Ha a megszakítási kérés prioritása nagyobb, mint az éppen futó programé, akkor az felfüggesztett állapotba kerül, és a →megszakításfeldolgozó szubrutin kapja meg a vezérlést. Ha a megszakítási kérés prioritása kisebb, mint az éppen aktív programé, akkor a kérés várakozási sorba kerül egy későbbi feldolgozás számára. Amikor egy megszakítás feldolgozása befejeződött, akkor a felfüggesztett programok és a várakozó megszakítási kérések közül a legnagyobb prioritással rendelkező kerül végrehajtásra.

megszakításkérés (interrupt request) – a számítógéprendszer valamely alegysége által jelzett igény a →központi vezérlőegység kisajátítására, valamely esemény felismerésére és feldolgozására. A ~ automatikusan az éppen feldolgozás alatt lévő program futásának felfüggesztéséhez és a speciális →megszakításfeldolgozó szubrutin indításához vezet (→megszakítás).

megszakításmaszkolás (interrupt masking) – a →központi vezérlőegység által végrehajtott olyan művelet, amelynek révén egy megszakítási szintet v. azok egy csoportját ideiglenesen kiiktatja a →megszakítási rendszerből. Az illető megszakításokat az →operációs rendszer rögzíti ugyan, de nem dolgozza fel rögtön. A →csapda jellegű megszakításokat rendszerint maga a felhasználó is maszkolhatja, de ilyenkor az eredmények helyességéről magának kell meggyőződnie.

megszakítható állapot (interruptable state) – a →központi vezérlőegység egyik lehetséges állapota: a →maszkolt megszakításokat kivéve bármely megszakítást azonnal figyelembe vesz és feldolgoz (többszintű megszakítási rendszerekben természetesen a →prioritásokat is figyelembe veszi).

megszakítható utasítás (interruptable instruction) – olyan →gépi utasítás, amelynél – ha megszakítás lép fel tényleges végrehajtása közben – a program felfüggeszthető; az utasítás végrehajtása a regiszterek tartalmának megőrzése mellett ott folytatódik, ahol abbamaradt (ezek az utasítások rendszerint

több memóriaciklust követelnek, s ezért megszakíthatóak).

megszakított program (interrupted program) – →multiprogramozásos rendszerekben végrehajtás alatt lévő olyan program, amely időlegesen elvesztette a →központi vezérlőegység felhasználási jogát, mivel →b/k művelet befejezésére vár. A ~ állapota abban különbözik a →felfüggesztett programétól, hogy még →prioritása sem adja meg a lehetőséget arra, hogy a központi vezérlőegység birtoklásáért versenyezzen mindaddig, amíg a kezdeményezett b/k művelet be nem fejeződött.

melegindítás (warm start) – →operációs rendszer újra munkába állítása olyan esetekben, amikor kezelési v. programozási hiba kivédésére a rendszer védelmi mechanizmusa aktivizálódik és a rendszer leáll. Általában nem igényel teljes rendszerbetöltést és →inicializálást, a hiba elhárítása után a rendszert újra lehet indítani (→hidegindítás).

mellékhatás (side effect) – valamely eljárás (program, programegység, utasítás stb.) olyan esetleges hatása, megnyilvánulása, amely a várt (programozott) hatás melléktermékeként érvényesül, s a programozási nyelveket megvalósító →fordítóprogramoknak tulajdonítható elsősorban. Jellegzetes és igen gyakori ~ az aritmetikai kifejezésekkel kapcsolatos: a legtöbb programozási nyelv nem határozza meg egy-egy operátor operandusainak kiértékelési sorrendjét. Így pl. az x=f()+g() kifejezésben, ahol v. az f(), v. a g() olyan változót (is) módosít, amelytől a másik is függ, az x értéke változhat aszerint, hogy az f() v. a g() került először kiértékelésre.

memória (memory) – számítógéprendszerek alapvető funkcionális egysége, amely információ – adatok és programok – tárolására alkalmas. A rendszerben elfoglalt helye, ill. funkciója alapján lehet: operatív ~, amely a végrehajtandó programot és a →központi egység által közvetlenül feldolgozható adatokat (→belső adatokat) tárolja, v. →háttértár, amely a későbbi feldolgozás (végrehajtás) céljára tárolt v. éppen egy előző feldolgozás, művelet eredményeként kapott programokat, ill. adatokat rögzíti. Az információk beírása a ~ba, ill. azok kiolvasása a →címezhetőség elve alapján történik.

A ~ alapvető jellemzői: →memóriakapacitás, →elérési idő, →hozzáférési módszer stb.

memóriaciklus (memory cycle) – 1. Azoknak a műveleteknek az összessége, amelyek egységnyi információnak az →operatív memóriába való beírásához, ill. az onnan történő kiolvasásához szükségesek. – 2. Az az időintervallum, amely az egységnyi információnak az operatív memóriába való beírásához, ill. az onnan történő kiolvasásához szükséges.

memóriagazdálkodás (memory management) – az →operációs rendszer egyik legfontosabb funkciója, megoldási módszere a különböző operációs rendszerek alapvető jellemzője. Feladata a memória kiosztása, állapotának állandó nyilvántartása, a szabad memóriazónák rendelkezésre bocsátása és kiosztása, a felszabaduló zónák felismerése és újrakiosztása, az elfoglalt zónák védelmének biztosítása stb. Legfontosabb stratégiái: →monoprogramozás, →multiprogramozás (rögzített és változó →partíciókkal), →virtuális memóriakezelés, →lapkezelés stb.

memóriahierarchia (memory hierarchy) – a különböző technológiával készült tárolóberendezéseknek (→memóriáknak) egy számítógéprendszeren belüli többszintes elrendezése. A hagyományosnak tekinthető ~ kétszintes: a felső szintet a félvezetős memória, az alsó szintet pedig a →mágneslemez alkotja. Nagyobb teljesítményű számítógépeknél az operatív memória is szintekre bontható. A felhasználó számára a ~ láthatatlan (transzparens), csak az egységes operatív memória létezik (→átfedési szerkezet, →virtuális memória, →lapkezelés stb.).

memóriakapacitás (memory capacity/size) – adott számítógéprendszer →operatív memóriájában tárolható információmennyiség alapvető jellemzője. A tárolórekeszek (→szavak, →bájtok) számával fejezhető ki.

memóriakiíratás (memory dump/printout) – az a művelet, amelynek révén az →operatív memória teljes v. részleges tartalmát valamely külső →adathordozóra (rendszerint →nyomtatóra) kimásolják. Jelentősége a programok belövési (tesztelési) fázisában van, amikor a hibás működés csupán a →forrásnyelvű program vizsgálatával nem deríthető fel. A ~t általában speciális rendszerprogram végzi el egy-egy végzetes hiba (ismeretlen utasítás-

kód, hibás cím, nem maszkolt →túlcsordulás stb.) után, de a felhasználónak lehetősége van bármilyen esetben kérni (a →munkavezérlő nyelv segítségével). Az egyetlen nyomozási lehetőség akkor, amikor magában az operációs rendszerben áll elő végzetes hiba.

memóriakiosztás (memory allocation) – 1. Az →operációs rendszer egyik fontos funkciója, amelynek célja többfelhasználós környezetben az operatív memória szétosztásával e fontos →erőforrás gazdaságos kihasználásának biztosítása (→memóriagazdálkodás). – 2. Programok →fordításának egyik alapvető fontosságú fázisa, amelynek során a programban definiált objektumokhoz a megfelelő nagyságú memóriazónákat hozzárendelik. A ~ módjának megfelelően erre három módszer ismeretes.

Statikus ~nál a memóriarekeszek kijelölése már a fordítási időben megtörténik, a memória lefoglaltsága a program végrehajtásának végéig változatlan marad. Ezt a fajta ~t valamennyi magas szintű programozási nyelv lehetővé teszi.

Az automatikus ~ a →blokkszerkezetű programozási nyelvekre jellemző (→ALGOL, →PL/1 stb.). A kívánt tárterület tényleges lefoglalására csak akkor kerül sor, amikor az a programblokk kerül végrehajtásra, amelyben deklarálták az illető adatokat (→lokális szimbólumokat). Amikor az eljárás végrehajtása befejeződött, a lefoglalt memóriarész felszabadul.

A vezérelt (dinamikus) ~ csak egyes magas szintű programozási nyelvekre érvényes (ilyenek pl. a →Pascal, a →Modula, az →Ada, a →CHILL stb.). A programozónak lehetősége van arra, hogy a végrehajtás alatt explicit módon vezérelhesse a ~t. Tetszés szerint blokkokat foglalhat le és szabadíthat fel az operatív memóriából úgy, ahogy a program pillanatnyi szükségletei megkívánják (→dinamikus változó).

memórialokáció (memory location) – valamely számítógép →operatív memóriájának olyan egységnyi része, amelynek tartalma egyetlen →memóriaciklus alatt olvasható v. írható felül, következésképpen egyedi címmel rendelkezik. A →bájtszervezésű számítógépeknél a ~ a bájt v. annak bizonyos páros számú többszöröse (2 bájt=félszó, 4 bájt=szó, 8 bájt=duplaszó). Nevezik még memóriarekesznek is.

memóriaszó (memory word) – az a maximális bitszámú vektor, amely egyetlen memóriaciklussal olvasható ki az operatív memóriából, vagyis egyetlen memória-hozzáféréssel érhető el. A számítógépek →architektúrájának egyik fontos jellemzője.

memóriatérkép (memory map) – 1. →Kötegelt feldolgozású rendszerekben az →operatív memória pillanatnyi állapotára, az egyidejűleg végrehajtás alatt lévő programokra vonatkozó információk összessége, amelyeket az operációs rendszer bocsát – kérésre – a gépkezelő rendelkezésére a vezérlőpult képernyőjén megjelenő táblázat formájában. A legfontosabb információk a →partíciók számára, méretére és foglaltsági állapotára, a →prioritásokra, a hozzárendelt →periferikus művekre stb. vonatkoznak. Ezek alapján a gépkezelő a pillanatnyi szükségleteknek megfelelően intézkedhet: új partíciókat hozhat létre, ill. szüntethet meg, prioritásokat változtathat meg stb. – 2. Átfedési szerkezettel rendelkező programoknál a párhuzamos ágak betöltésére vonatkozó információk.

memóriavédelem (memory protection) – az a (→hardver v. →szoftver megoldású) mechanizmus, amelynek révén az →operatív memória egy részén tárolt információ megvédelmezhető az avatatlan hozzáféréstől. A ~ elengedhetetlenül szükséges a →multiprogramozású rendszerekben, mert csak így akadályozható meg, hogy egyik program (logikai hiba miatt) felülírjon egy másik programhoz rendelt memóriarészt (→partíciót). A ~nek különböző szintjei lehetnek: védelem felülírás ellen, olvasás ellen, írás és olvasás ellen.

menü (menu) – →menüvezérelt üzemmódban dolgozó programtermékek funkcióválasztékának megjelenítése a képernyős →terminálon. Lényegében egy v. több →ablak, amelyben a választható funkciók listája szerepel. Komplexebb programtermékek a ~k egész hierarchiáját nyújtják, ahol a fő~ a legfontosabb funkciókat tartalmazza, ezek kiválasztása pedig automatikusan egy alsóbb szintű ~t aktivizál. Visszatérni egy felsőbb szintre rendszerint →funkcionális billentyű segítségével lehet (általában Esc).

menüvezérelt üzemmód (menu-driven mode) – a →párbeszédes üzemmód modern, felhasználóbarát környezetet biztosító változata. Lényege, hogy a végrehajtás alatt álló programtermékek funkcióit a →képernyő →ablakaiban kivetítve, →menük formájában bocsátják a felhasználó rendelkezésére. A felkínált lehetőségek közül a felhasználó többféleképpen választhat: 1) a →helyőr megfelelő mozgatásával (→nyílbillentyű, →egér, →botkormány stb.); 2) az egyes funkciókhoz rendelt karakteres v. →funkcionális billentyű leütésével; 3) érintésérzékeny képernyőknél érintéssel v. →fényceruza segítségével.

merevlemez (hard disk) – a →hajlékonylemez megjelenése óta az eredeti →mágneslemez megnevezése.

mesterséges intelligencia (artificial intelligence) – minden olyan rendszer (esetleg számítógépen megvalósított algoritmus), amely az emberi magatartás modellezésére alkalmas. A kutatási irányzatok egyik ága olyan rendszerek előállítására irányul, amelyek intelligens műveletek végzésére használhatók. Ezek elsőrendű célja nem az emberre jellemző módszerek, eljárások hű másolása, hanem bizonyos emberi tevékenységek egyszerű, gyors és olcsó elvégzése. Ide tartozik az egyszerű robotok előállítása, a természetes nyelvek számítógépes fordítása stb. A másik irányzat az emberi intelligencia modellezése tanulmányozás céljából. Ide tartozik a matematikai tételek bizonyítása, a döntéselmélet kidolgozása, a tanulás mechanizmusának tanulmányozása, a sakkjáték számítógépes modellezése stb.

A ~ kutatásának kezdetét a hetvenes években Winograd kérdés-válasz rendszere jelentette. Azóta a kutatás központi feladataként a tudás ábrázolása és hasznosíthatósága került előtérbe. Számítástechnikai vonatkozásban az új elvű, logikai programozás tekinthető sokat ígérő eredménynek.

mesterséges nyelv (artificial language) – az a nyelv, amely nem természetes úton alakult ki, hanem az emberi tevékenység valamely területén az információcsere megvalósítására mesterséges úton hoztak létre. Ilyen megközelítésben ~ az eszperantó, az összes →programozási nyelv, de annak tekinthető a →blokkdiagram, a →döntési táblázat stb. is.

A ~eket jellemzi, hogy sokkal kevesebb mondatformával rendelkeznek, mint egy természetes nyelv, mondataik (majdnem) kétértelműség-mentesek, mivel aránylag kevés számú, pontos szintaktikai szabály alapján épülnek fel.

metakarakter (metacharacter) – valamely →metanyelv →terminális szimbóluma, vagyis olyan szimbólum, amely nem tartozik a leírandó nyelv szimbólumai közé, csupán leíró szerepe van. Pl. a →Backus-Naur Forma ~ei: <, >, ::=,| stb. A ::= ~, jelentése pedig az, hogy a bal oldalán álló →metaváltozó helyettesíthető a jobb oldalán álló →metakifejezés bármelyik alternatívájával. A ::= szimbólum a metanyelv saját szimbóluma, amelyről kijelentik, hogy nem tartozik a definiálandó nyelv ábécéjébe.

metakifejezés (metaexpression) – amely →metanyelv szabályainak jobb oldalán álló, →metakonstansokból, →metaváltozókból és →metakarakterekből felépített kifejezés. A programozási nyelvekben az előjel nélküli egész számot pl. rendszerint a következő szabály írja le:

<előjel–n.–egész>::= <számjegy> |
<előjel–n.–egész> <számjegy>

vagyis előjel nélküli egész szám lehet egy önmagában álló számjegy, v. egy olyan, előjel nélküli egész szám, amelyet egy újabb számjegy követ. A gyakorlatban az ilyen rekurzív meghatározásokat rendszerint elkerülik, újabb metakarakterek bevezetésével a ~t átalakítják. Ha $ jelentése az utána elhelyezett metakonstans v. metaváltozó esetleg üres sorozata, akkor az előjel nélküli egész számot már rekurzivitás nélküli új ~ határozhatja meg:

<előjel–n.–egész>::=<számjegy>$
<számjegy>

vagyis előjel nélküli egész lehet v. egy egymagában álló szám v. ilyen számok végtelen sorozata.

A →FORTRAN programozási nyelv →címkéinek meghatározását a következő ~ határozhatja meg:

<címke>::=<számjegy>$
{<számjegy>}0/4

vagyis maximálisan 5 számjegyű előjel nélküli egész szám lehet címke ebben a nyelvben. A

szabály jobb oldalán álló ~ben a <, >, {, }, 0, /, 4 metakarakterek.

metakonstans (metaconstant) – valamely →metanyelv által felhasznált konstans, amelyet adott programozási nyelv szintaktikai struktúráinak leírására használnak. Pl. a →Backus-Naur Forma segítségével definiálandó nyelvtan →terminális szimbólumai az előbbi ~ai.

metanyelv (metalanguage) – fogalmát N. Chomsky vezette be az élő nyelvek →szintaxisának leírására; megfogalmazása szerint: más nyelvek leírására használható nyelv v. formalizmus. Általános jellegű szakkifejezés, ugyanis valamely természetes nyelvtől kezdve bármilyen szimbolikus jelölésmód alkothat ~et. A magas szintű programozási nyelvek →konkrét szintaxisának leírására leggyakrabban használt ~ a →Backus-Naur Forma.

metaszimbólum (metasymbol) – →metakarakter

metaváltozó (metavariable) – valamely →metanyelv által felhasznált változó, amelyet adott programozási nyelv szintaktikai struktúráinak leírására használnak. A →Backus-Naur Forma segítségével definiálandó nyelvtan →nemterminális szimbólumai az előbbi ~i.

Pl. a programozási nyelvekben a <számjegy> ~, meghatározását a következő szabály adja:

<számjegy>::=0|1|2|3|4|5|6|7|8|9

jelentése pedig: (decimális) számjegy a 0 v. 1 v. 2 v. ... v. 9 szimbólumok bármelyike lehet (a tíz decimális számjegy itt →metakonstans és egyben a leírt programozási nyelv →terminális szimbóluma).

mező (field) – →adatstruktúrák felépítésében részt vevő legkisebb címezhető egység. Alapvető jellemzői: a szelektoreljárás, vagyis az az algoritmus, amelynek révén a ~ lokalizálható, ill. kiemelhető az adatstruktúrából, és a ~ típusa, vagyis a ~höz rendelhető értékek típusa.

mikroművelet (microoperation) – egy →mikroutasítás végrehajtásával kifejtett elemi, funkcionális művelet. Általánosan használt ~ek: a bitszintű összeadás, kivonás, logikai művelet, a léptetések, az átvitelek stb.

mikroprocesszor (microprocessor) – digitális integrált áramköri alkatrész, amely egy számítógéprendszer →központi vezérlőegységének megfelelő funkciókat lát el. A világon az első ~t 1971-ben az INTEL cég jelentette meg 4004-es típusjelzéssel, s azóta több generáció is napvilágot látott. A technológia fejlődése lehetővé tette nem csupán egy központi vezérlőegység, hanem a teljes számítógépmag (központi vezérlőegység, memória, be-, ill. kimeneti →interfész) egyetlen tok formájában való megvalósítását, és így megszületett az egylapkás →mikroszámítógép. A fejlődés mennyiségi és minőségi fokmérője a ~ által közvetlenül feldolgozható →bitek száma. Az első ~ 4 bites volt, majd nagyszámú 8 bites típust dolgoztak ki (I8080, M6800, Z80 stb.); ezt követően jelentek meg a 16, majd a 32 bites ~ok (I8086, I80186, I80286, I80386, I80486 stb.).

mikroprogram (microprogram) – 1. →Mikroprogramozott vezérlést megvalósító, →permanens memóriában tárolt →mikroutasítások sorozata. – 2. →Mikroprocesszoros rendszerek vezérlését biztosító →program. Egy ideig majdnem kizárólag a →gépi kóddal megegyező, géptől függő →asszembler nyelven készült. Később a számítógépes programfejlesztés tapasztalatai alapján kialakultak a magas szintű nyelven történő programozás lehetőségei is (pl. →PL/M, →Pascal stb.). A →fordítást végző programok futtathatók fejlesztőgépeken (→keresztszoftver), v. az adott alkalmazói rendszereknek lehetnek saját fejlesztő programjai.

mikroprogramozott logika (microprogrammed logic) – →mikroprocesszorokkal kapcsolatos fogalom, amely a hagyományos logikai rendszereknek mikroprocesszoros rendszerekkel történő helyettesítésére vonatkozik. A helyettesítés v. a logikai elemekre v. az egész feladatra elvégezhető. Az alapvető különbség az, hogy a digitális logikai áramkörök alkalmazásánál a különböző logikai változók és állapotok párhuzamos ágakon haladhatnak és csomópontokban kerülnek kiértékelésre, míg a mikroprocesszor ugyanazokat az eredményeket egy program utasításainak egymás utáni végrehajtásával állítja elő. A ~ több nagyságrenddel kisebb sebességet biztosít, ezért a gyakorlatban sohasem az egyes digitális áramkörökkel megvalósított rendszert helyettesítik a ~val, hanem az alap-

feladatot oldják meg mikroprocesszoros rendszer és a hozzá tartozó program segítségével. A legtöbb felhasználásnál követelmény, hogy bekapcsolás után a ~ azonnal működőképes legyen. Ez csak →permanens memóriákba égetett vezérlőprogramokkal valósítható meg.

mikroprogramozott számítógép (microprogrammed computer) – olyan →digitális számítógép, amelynél a →központi vezérlőegység megvalósítására →mikroprogramozott vezérlést használtak. Az első, mai értelemben vett ~ az 1964-ben megjelent IBM System/360 számítógépcsalád.

mikroprogramozott vezérlés (microprogrammed control) – M. V. Wilkes által javasolt módszer a →központi vezérlőegység megvalósítására, hogy az időzítő-vezérlő jeleket az adott utasítás hatására aktivizált →mikroprogram állítsa elő. A mikroprogram →mikroutasítások sorozata, és a mikroprogramtárban található. A ~ kevésbé gyors ugyan, mint a huzalozott vezérlés, de a vezérlőegység egyszerűbb, és a vezérlési lehetőségek széles határok között változtathatók (pl. az utasításkészlet könnyen bővíthető v. módosítható).

mikroszámítógép (microcomputer) – valamely →mikroprocesszor köré épített rendszer, vagyis olyan számítógép, amelynél a →központi vezérlőegység feladatát mikroprocesszor látja el. A ~ többi funkcionális elemét a következők alkotják: a változatlan adatok, ill. alapprogramok (pl. →BIOS) tárolására szolgáló →permanens memóriák (ROM), a változó adatok, ill. felhasználói programok rögzítésére használt →véletlen hozzáférésű memóriák (RAM), a belső adat- és vezérlőjel-forgalmat lebonyolító sínrendszer és a külvilággal kapcsolatot teremtő →interfészek.

mikroutasítás (microinstruction) – olyan bitvektor, amely egy →processzor állapotváltozásához szükséges vezérlőjeleket szolgáltatja (→mikroprogramozott vezérlés). Felépítése az adott rendszerre jellemző, egyben az →architektúra kialakítását is meghatározza. Egyetlen ~ általában a →hardver erőforrások különböző csoportjait is vezérli, kihasználva a párhuzamos végrehajtásból származó lehetőségeket.

A horizontális felépítésű ~ban számos párhuzamos végrehajtást vezérlő mező van; a vezérlőjelek közvetlen generálására vezérlőbiteket használ. A vertikális felépítésű ~ban egyetlen →mikroművelet van; a hagyományos utasításformátum műveleti kódját és címmezőjét használja.

minimálsúlyozású adatútképzés (minimum weight routing) – olyan →adatútképzés, amely minimalizálja az útvonalban alkalmazott összeköttetések súlyainak összegét. Ezek a súlyozások a kapcsolópontoknál fellépő késések, költség- v. hibamértékek lehetnek, azaz bármi, ami az átvitel folyamán halmozódik és így minimalizálni kell.

minimax-tétel (minimax principle) – a →játékelmélet (Neumann Jánostól származó) elve, ill. kritériuma, amely szerint a ~en alapuló stratégia alkalmazása a játékosoknak bármilyen ellenjátéknál a legkisebb veszteséggel a legnagyobb nyereséget biztosítja. Kimondja, hogy a kétszemélyes zéróösszegű véges játéknak mindig van megoldása, és az egyik játékos minimális nyereségeinek maximuma egyenlő a másik játékos maximális veszteségeinek minimumával. Döntési feladatokra alkalmazva: annak a stratégiának a követése, amely a maximális kockázat lehetőségét minimalizálja. Függvények megközelítésére alkalmazva: annak a formulának a felhasználása, amely a maximális hiba lehetőségét minimalizálja.

minősített kifejezés (qualified expression) – →erősen típusos nyelvekben (pl. →Ada) használható olyan kifejezés, amelynek →[adat]típusát explicit módon feltüntették. Jelentősége, hogy egyes kétértelműségeket eleve megszüntet, s ezáltal megkönnyíti a →fordítóprogram által végzett ellenőrzéseket.

minősített név (qualified name) – magas szintű →programozási nyelvekben (pl. →COBOL) használatos →azonosító, amelyre a hivatkozás úgy válik egyértelművé, hogy egy v. több magasabb szintű névvel (→minősítővel) használják. A minősítőkből annyit kell a minősítésre használni, hogy a hivatkozás egyértelmű legyen; nem szükséges a hierarchia minden közbenső szintjét felhasználni, ha az egyértelműség enélkül is biztosítva van.

minősítő (qualifier) – hierarchikus struktúrákat megengedő, magas szintű →programozási nyelvekben (pl. →COBOL) használatos név, amely alacsonyabb szintű, nem

egyértelmű adatot egyértelművé tesz (→minősített név).

mintaillesztés (pattern matching) – a →logikai programozás egyik alapvető jellemzője, módszere. Lehetővé teszi, hogy először általános programsémákat (programmegoldási sémákat) fogalmazzanak meg, majd a konkrét feladat programozásakor csak magát a feladatot írják le, s ahhoz keressék ki a megfelelő sémát. Sikertelen próbálkozások után →visszalépegetéssel próbálható ki az adott feladatra vonatkoztatható többi séma is, amíg v. megoldódik a feladat v. bizonyíthatóan kimutatható az ilyen séma hiánya (→Prolog).

MIPS (Million Instruction Per Second) – →központi vezérlőegységek műveleti sebességének mértékegysége (millió utasítás másodpercenként).

mix (mix) – általában súlyozott összeg, amelynek tagjai a számítógép v. program jellemző értékeit (pl. végrehajtási időket) tényezőként tartalmazó szorzatok. Rendszerint számítógéprendszerek hatékonyságának közelítő becslésére, jellemzésére használják (ilyen pl. a Gibson-mix index).

mnemonikus kód (mnemonic code) – a →gépi utasítások és a magas szintű →programozási nyelvek utasításainak könnyebb megjegyzése és használata céljából alkalmazott szimbolikus jelölésmód, amely (legtöbbször angol) szavak v. rövidítések útján utal az elvégzendő művelet jellegére és típusára. A ~ az →asszembler nyelvektől kezdve (pl. AD2=add halfword, MPD=multipy decimal stb.) egészen a magas szintű programozási nyelvekig (pl. ADD, DO, COMPUTE, FOR, INSPECT stb.) mindenhol megtalálható.

modell (model) – természeti és társadalmi jelenségek olyan leírása, amelyben fizikai elemeket gondolkodási elemek (logika, matematikai formalizmus stb.) helyettesítenek az illető jelenség tanulmányozása céljából. Számítástechnikai szempontból különösen fontosak a matematikai ~ek és azok, amelyek a →programozási nyelvek elemeivel írhatók le. Egy test mozgásának matematikai ~jeként vizsgálható pl. az a differenciálegyenlet, amellyel az illető mozgás leírható. Ha a differenciálegyenlet elég jó megközelítéssel írja le a mozgást, akkor a fizikai jelenség tanulmányozására jól felhasználható. A ~ legfőbb tudományos funkciója tulajdonképpen nem maga a leírás, hanem a jelenségek magyarázata. Alapvető formái a →leíró modell és a →magyarázó modell.

modem (modem) – a modulátor-demodulátor rövidítéséből alkotott neve annak az →adatátviteli hálózatokban (is) használatos elektronikus készüléknek, amely lehetővé teszi az analóg csatornáknak digitális információk továbbítására való felhasználását. Az adóállomás ~ segítségével csatlakozik az átviteli vonalra, amely moduláció révén előállítja az átvitelre alkalmas analóg jeleket. A vevő ugyancsak ~ útján kapja a demoduláció révén visszanyert digitális információt.

módosított komplemens kód (modified complement code) – a →komplemens kóddal egyenértékű – azonos információtartalmú -, de annál eggyel több bitet tartalmazó ábrázolás. A kiegészítő bit a komplemens kód legnagyobb helyértékű bitpozíciójának tartalmát ismétli. A ~ előnye additív műveleteknél a →túlcsordulás vizsgálata során jelentkezik.

módszerhiba (method error) – valamely eljárás számítógépre vitele során elkövethető hiba. Származhat az illető feladat megoldására hibásan kiválasztott v. megtervezett algoritmus felhasználásából, a feladat nem kielégítő ismeretéből, a lehetséges állapotok (főleg a határállapotok) figyelmen kívül hagyásából, a →hibakorlátok helytelen megválasztásából stb.

modul (module) – programozástechnikai értelmezésben utasítások sorozatára épülő, önálló névvel rendelkező, önmagában érthető programegység. Ebben az értelemben ~ pl. egy →FORTRAN szubrutin, egy →asszembler nyelvű szekció stb. A →moduláris programozás módszertana szempontjából a ~ meghatározását még ki lehet egészíteni a következő megállapításokkal: a ~ a program által megoldott feladatnak egyik →logikai egysége; a fejlesztés szempontjából önállóan tervezhető, kódolható és tesztelhető programegység.

A ~ok osztályozásának szempontjai: a funkció, a működési logika és a más ~okhoz fűződő kapcsolat. A ~ funkciója az az átalakítás, amelyet a bemenő adatokon elvégez. Ilyen értelmezésben csak azt tartják szem előtt, hogy mit csinál, és nem érdekes, hogy

hogyan működik. Működési logikán azt értik, hogy a ~ miként hajtja végre az adott funkciót.

A ~ok – funkciójuk szerinti – osztályozása 4 típusú programegységet tart nyilván: →adatmodulokat, →eljárásmodulokat, →vezérlőmodulokat és →b/k modulokat. A tervezésben igen fontos szempont, hogy a ~ok minél függetlenebbek legyenek egymástól (→moduláris programozás).

Modula – →konkurens folyamatok programozására N. Wirth által kifejlesztett magas szintű →programozási nyelv. Első változata tkp. csak a konkurens elvű programozás oktatására készült, így szekvenciális része a →Pascalnak csak részhalmaza. Elsőrendű célkitűzése a párhuzamos, egymással →versenyhelyzetben álló folyamatok programozása. Az →erősen típusos nyelvek családjába tartozik, s hatékony eszközöket tartalmaz új →[adat]típusok előállítására. Lehetővé teszi a →külön fordítást. A legtöbb jellegzetes →operációs rendszeri funkciót ellátja, így különösen alkalmas kis méretű, →valós idejű rendszer előállítására. A ~ sikere láttán N. Wirth kidolgozta annak fejlettebb változatát is, Modula-2 néven, amelyben jobban figyelembe vette a gyakorlati szempontokat is.

A ~ egyprocesszoros rendszerek fejlesztésére készült, gyakorlatilag nem használható osztott architektúráknál. Eszközeinek igen nagy részét örökölték a legkorszerűbbnek tartott programozási nyelvek, az →Ada és a →CHILL.

moduláris programozás (modular programming) – korszerű programfejlesztési módszertan, amely az építőkockák elvén alapszik. Alapelve az, hogy a számítógéppel megoldandó feladatot egymástól független részfeladatokra osztják fel, amelyeket különkülön lehet megírni (esetleg átvenni már meglévő programtermékekből), s főleg egyenként lehet tesztelni, amikor a főhangsúly a programfunkció helyes működésének elérésére helyezhető. A →modul kipróbálásakor azt kell igazolni, hogy az előírt formában megadott bemenő adatain az előírt átalakításokat helyesen, teljes egészében végrehajtja. A programtermék megvalósításának második fázisában a már kész részprogramokból (modulokból) állítják össze a végső programot. Ebben a fázisban a tesztelés is új jelleget ölt, a figyelem a program egységes logikai működésére összpontosul. A programtermék kipróbálásakor a modulok közötti információcseréről és a vezérlésátadásokról kell kimutatni, hogy azok a programspecifikációkban megadott módon történtek meg.

A ~ egyik leghasznosabb újítása, hogy elkerülhetővé teszi bizonyos tevékenységek sokszori újraprogramozását. A harmadik generációs →operációs rendszerek megjelenése óta lehetőség van a már tesztelt programegységek elhelyezésére →modulkönyvtárakban, biztosítva ezáltal azoknak utólagos felhasználását. Mivel a modul egyszerű, kis logikai egységet képvisel, a programozó könnyen alkalmazhatja a programozási előírásokat, megfelelően választhatja meg a programozási nyelvet, hiánytalan, pontos dokumentációt készíthet számára. A modul viszonylag rövid idő alatt alaposan tesztelhető, ami biztosíték arra, hogy minden programozási egység jó minőségű, megbízható építőegység legyen a programtermék felépítésében, s ezáltal elérhető a végső termék magas fokú megbízhatósága.

modulkönyvtár (module library) – olyan →könyvtár, amelybe →fordítás útján nyert →tárgymodulokat rögzítenek. A →kapcsolóprogramok túlnyomó többsége közvetlenül a ~ból nyerheti a kapcsolandó programegységeket. Tekintettel arra, hogy az →operációs rendszerekben a tárgymodulok szerkezete és kódolása szabványosított, a ~ba bármilyen programozási (→asszembler, ill. magas szintű) nyelv fordítóprogramja rögzítheti az előállított modulokat. Ugyancsak a szabványosításnak köszönhetően a kapcsolóprogram is egységes programmá építheti fel az eredetileg különböző forrásnyelvű programegységeket, ami a →moduláris programozás egyik alapvető feltétele.

modulo-N összeadás (modulo-N addition) – a nullára és a természetes számokra értelmezett művelet: két szám ~i eredménye a számok algebrai összegének az N számmal való osztása útján nyert maradék, vagyis

$$\mathrm{mod}\,N\,(a+b) = a+b-p\cdot N \quad \text{ha } p\cdot N < a+b < (p+1)\cdot N$$

mondat (sentence) – 1. Valamely (élő v. programozási) nyelv szógyűjteményében

szereplő szavakból összeállítható szerkezeti forma. Az illető nyelv szempontjából jól meghatározott jelentése, →szemantikája van. A ~ok formáját meghatározó szabályok összessége a →szintaxis, a ~okhoz rendelhető jelentés v. értelmezés a szemantika. – 2. A →generatív nyelvtan szempontjából olyan →mondatforma, amely csak →terminális szimbólumokból épül fel.

mondatforma (sentential form) – a →generatív nyelvtannal kapcsolatos fogalom. →Terminális és/vagy →nemterminális szimbólumok olyan sorozata, amely az illető nyelvtan →helyettesítési szabályainak a →kezdőszimbólumra való ismételt alkalmazásával állítható elő. A csak terminális szimbólumokból álló ~t mondatnak nevezik. Egy megadott nyelvtan segítségével előállítható mondatok összessége az adott nyelvtan által meghatározott →nyelv.

mondatszerkezetű nyelvtan (phrase structure grammar) – a →Chomsky-hierarchia osztályozása szerint 0 típusú nyelvtan. A legáltalánosabb típusú nyelvtan, →helyettesítési szabályaira semmilyen megszorítás nem vonatkozik. A szabályok bal oldalán →terminális és →nemterminális szimbólumokból alkotott tetszőleges nem üres szöveg áll, míg a jobb oldalon megengedett ezen felül az üres szöveg is.

monitor (monitor) – 1. Az →operációs rendszer részét alkotó vezérlőprogramok összessége. Legfontosabb funkciói: a rendszer folytonos működésének biztosítása, az →erőforrásokkal való gazdálkodás optimalizálása, a →multiprogramozás támogatása.

A ~ rendszerint két alapvető részre oszlik. Egyik a rezidens rész, amely a legfontosabb, leggyakrabban használt vezérlőprogramokat tartalmazza, s állandó jelleggel az →operatív memóriában tartózkodik. A másik rész a ritkábban használt vezérlőprogramokat tartalmazza, s rendszerlemezen (v. rendszerszalagon) található, ahonnan csak tényleges felhasználásuk előtt kerülnek a vezérlőprogramok az operatív memóriába.

Funkcionális szempontból a következő legfontosabb részekből épül fel: →felügyelőprogram (→megszakítási rendszer, →prioritások, →b/k műveletek stb. vezérlésére); →központi monitor (a →hardver és az operációs rendszer többi része közötti logikai kapcsolat megteremtésére); →[adat]állomány-kezelő rendszer (a →logikai [adat]állományokkal kapcsolatos vezérlési és feldolgozási műveletek elvégzésére); a →kötegelt feldolgozásnál még ide tartozik a →munkaütemező (a →munkák automatikus láncolásának és végrehajtásának ütemezésére). – 2. →Megjelenítő. – 3. Mikroszámítógépeknél →hibakereső és →nyomkövetést végző program. – 4. Hálózatok pillanatnyi állapotát mutató programok.

monokróm képernyőadapter (monochrome display adapter) – IBM →személyi számítógépekhez kifejlesztett →képernyőadapter. Csak →szöveges módban működik, és a tárolt alfanumerikus és grafikus karaktereket csak egy színben képes megjeleníteni. A nagyobb →felbontás (720 x 350 →képpont) nagyon jól olvasható szövegmegjelenítést biztosít, de nélkülözi a grafikát és a színeket. A jobb grafikai lehetőségek biztosítására létrehozták a ~ módosított változatait, amelyek közül a Hercules a legismertebb. Ez egyesíti a grafikai lehetőségeket a ~ jó minőségű szövegmegjelenítésével.

monoprogramozás (monoprogramming) – a →memóriagazdálkodás legegyszerűbb stratégiája, amely lehetőséget ad a →munkák automatikus, egymás utáni végrehajtására → kötegelt feldolgozású környezetben. Adott pillanatban az operatív memóriában a →felügyelőprogramon kívül egyetlen felhasználói program van jelen, amely korlátlanul rendelkezik a rendszer összes →erőforrásával. A módszer előnye, hogy az automatikus kötegelt feldolgozást igen jó hatásfokkal valósítja meg, hátránya viszont, hogy meg sem kísérli a rendszer átbocsátóképességét (és ezzel együtt a válaszidőt) optimalizálni (egyes számítások szerint a →központi vezérlőegység kihasználtsági foka nem haladja meg a 25%-ot). Ma már csak speciális célú rendszerekben és →személyi számítógépeknél használják, a kötegelt feldolgozás másik formájával szemben (→multiprogramozás).

Monte-Carlo módszer (Monte-Carlo method) – sztochasztikus szimuláción v. sztochasztikus modellezésen alapuló numerikus módszer. Lényege az, hogy a megoldandó feladathoz olyan sztochasztikus modellt szer-

kesztenek, amelynek sokszori lejátszása mellett a keresett számra v. a függvény bizonyos jellemzőire jó statisztikus becslést kapnak. A modell lejátszásához véletlen számokból indulnak ki, amelyeket valamilyen fizikai megoldásból, véletlenszám-táblázatból v. pszeudovéletlen számokat generáló módszer segítségével nyerhetnek. Miután elkészült a vizsgálandó jelenség matematikai modellje, a megfelelő véletlen számokkal többször lejátsszák. A ~ használata akkor célszerű, ha egyéb numerikus módszerek nem felelnek meg v. túlságosan bonyolultak, és nincs szükség túl nagy pontosságra.

morzsa (chip) – egyetlen félvezető lapkán létrehozott olyan funkcionális →integrált áramkör, amelyen nemcsak az alkatrészek közötti, hanem a funkcionális egységek közötti összeköttetéseket is „tokon belül" hozták létre, ami a megbízhatóság jelentős növekedését eredményezi. Jellegzetes képviselője a →mikroprocesszor, amely egy számítógép →központi vezérlőegységének megfelelő funkciókat lát el.

mozgató mód (move mode) – →logikai [adat]állományok feldolgozásában használatos módszer a →logikai rekordok elérésére. Lényege az, hogy az olvasóutasítás kiadásakor az →operációs rendszer a bemenő →pufferből átírja a címzett logikai rekordot az annak fenntartott memóriaterületre, ill. íróutasításnál a kiírandó logikai rekordot átírja a kimeneti puffer meghatározott (első szabad) helyére. Ez azt jelenti, hogy a felhasználó teljesen figyelmen kívül hagyhatja a rekordok →blokkolásával kapcsolatos műveleteket, csakis a logikai rekord szintjén dolgozik a feldolgozandó adatszerkezetekkel (szemben a →helymeghatározó móddal).

multiplexcsatorna (multiplexer channel) – olyan →adatcsatorna, amely az →időmultiplexelés elve alapján több be-, ill. kimeneti készüléket tud egyszerre működtetni. Valójában több alcsatornát tartalmaz, s azok mindegyike ismeri az összes információt ahhoz, hogy valamelyik készüléket vezérelje. Multiplex üzemmódban az egyes készülékek logikailag csak annyi időre kapcsolódnak a csatornához, amennyi az egységnyi adatmennyiség (egy →bájt, bizonyos számú bájt, esetleg blokk) átviteléhez szükséges. A ~hoz általában a viszonylag lassú készülékeket csatlakoztatják, mint pl. a →nyomtatót, a →kártyaolvasót stb. Bizonyos esetekben a ~ működtethet egyetlen készüléket is, de ezt nagy sebességgel (szelektor v. lökésszerű üzemmód). Ilyenkor az időmultiplex üzemmódot meghatározott feltételek felfüggesztik, és a ~ →szelektorcsatornaként működik.

multiplexelés (multiplexing) – általános értelmezésben egy →erőforrás több felhasználó számára történő olyan kiosztása, hogy a felhasználónál azt a látszatot keltse: az illető erőforrás állandóan rendelkezésére áll. Számítástechnikai értelmezésben olyan eljárás, amelynek során egy nagyon gyorsan működő berendezés több, lassan működő berendezést látszólag párhuzamosan szolgál ki. A számítógéprendszerekben a legismertebb ilyen eljárás az →időosztásos multiplexelés, amelynél a gyors berendezés (→multiplexcsatorna) kis időközönként, sorrendben egymás után szolgál ki több lassú berendezést. A távolsági adatátvitelben használatos legismertebb eljárás a frekvenciaosztásos multiplexelés, amelynél a nagy sebességű átviteli csatorna frekvenciasávja van felosztva a lassú terminálok kiszolgálására. Tkp. ~t használ az időosztásos üzemmódban dolgozó →operációs rendszer is, amikor a →központi vezérlőegységet bocsátja meghatározott időközönként az egyszerre végrehajtás alatt lévő programok rendelkezésére.

multiplexer (multiplexer) – helyi v. távolsági adatátvitel végrehajtására használatos olyan funkcionális berendezés, amely a →multiplexelést megvalósítja.

multiprocesszálás (multiprocessing) – a →multiprogramozásnak két v. több →processzorral történő megvalósítása. A különböző feladatoknak egy programon belüli párhuzamos végrehajtásához speciális programozási nyelvek szükségesek, amelyek a következő legfontosabb lehetőségekkel kell hogy rendelkezzenek: az egyes részfeladatok előállítása, ill. megszüntetése, ezek végrehajtásának indítása, ill. felfüggesztése, az egyes részfeladatokhoz tartozó adatzónák védelme, a különböző részfeladatok végrehajtásának szinkronizálása stb.

multiprocesszor (multiprocessor) – olyan számítógéprendszer, amelyben két v. több

→processzor van egyetlen →központi vezérlőegységbe összeépítve (leggyakoribb a biprocesszor, vagyis a két processzort tartalmazó rendszer). A processzorok együttműködését speciális →operációs rendszer teszi lehetővé. A –os rendszer alapvető jellemvonása, hogy egyetlen →operatív memóriával rendelkezik (szemben a →többprocesszoros rendszerekkel, amelyeknél szintén több processzor működik közre, de mindegyiknek saját operatív memóriával van kapcsolata).

multiprogramozás (multiprogramming) – több felhasználót kiszolgáló, egyetlen →központi egységet tartalmazó számítógéprendszereknél az a lehetőség, hogy egyidejűleg több program van jelen az →operatív memóriában, s ezek látszólag egyszerre vannak végrehajtás alatt. A – alapelve az, hogy a →központi vezérlőegységet annak a programnak utalják ki, amely adott pillanatban azt fel is tudja használni. Pl. bizonyos program, amely a →háttértárral történő adatcsere befejezésére vár, feleslegesen tartaná lefoglalva a központi vezérlőegységet, s így azt más program részére lehet kiutalni. Ha egyszerre több program is készen áll a központi vezérlőegység felhasználására, fontossági sorrendben választják ki azt, amelynek végrehajtását folytatni kell (→prioritás). A – megvalósítására az operációs rendszernek speciális szubrutinokkal kell rendelkeznie, amelyek közül kulcsszerepet játszik a →felügyelőprogram és az →ütemezőprogram.

munka (job) – →kötegelt feldolgozásnál egy feladat számítógépes megoldásában részt vevő programok és az azok által igényelt →erőforrások összességére vonatkozó szakkifejezés. Dinamikus megközelítésben a – meghatározott típusú programok meghatározott sorrendű, láncolt végrehajtása. A →szoftver és →hardver erőforrások kiutalása az →operációs rendszer munkakezelő része által értelmezett és végrehajtott parancsok révén történik. Ilyen alapon az egész –, mint egység, jól elkülöníthető fázisokra tagolódik (→fordítás, →kapcsolás, →betöltés, erőforrás-foglalás, végrehajtás, kezelőprogramok és rendszerprogramok végrehajtása stb).

Statikus szempontból a – a →parancsnyelv szabályai alapján kódolt utasítások sorozata, amellyel a felhasználó a szükséges szoftver és hardver erőforrások kiutalását igényli. Az utasítássorozatot két speciális parancs határolja fizikailag: a – kezdetét és végét jelző utasítás (JOB és END OF JOB). A –t meghatározó →vezérkártyákat rendszerint a standard →bemeneti egységen keresztül juttatják a kiutalt →partícióba, de indítható – →katalogizált eljárás v. →gépkezelői beavatkozás útján is.

munka-[adat]állomány (job file) – az az →[adat]állomány, amely a feldolgozás közben időlegesen tárolásra szánt részeredményeket tartalmazza, s csak az adott munkafázis tartamára létezik. Különleges jelentősége van azoknak a –oknak, amelyeket az →operációs rendszer saját szükségletei számára állít elő. Ilyen –ra kerülnek pl. a több menetben dolgozó →fordítóprogramok közbenső kódjai. –okra van szükség a →külső rendezés, ill. az →összeválogatás megvalósítására, és használhatnak ilyen [adat]állományokat a felhasználói programok is.

munkaállomás (workstation) – irodai, ipari stb. munkahelyeken, rendszerint →mikroszámítógép köré épített, jól meghatározott feladatokat elvégző eszközök csoportja. Minimális kiépítése a mikroszámítógépen kívül legalább egy képernyős →megjelenítőt, egy →billentyűzetet és egy →hajlékonylemezes meghajtót tartalmaz, de használhat →merevlemezes meghajtót, →nyomtatót, esetleg a munkahelyre jellemző speciális →periferikus műveket is (pl. →optikai karakterolvasót, vonalkódolvasót, grafikus adatok bevitelére és kivitelére alkalmas eszközöket stb.). A –ok →helyi hálózatokba is köthetők.

munkahalmaz (working set) – →lapkezeléssel megvalósított többfelhasználós környezetben a programok viselkedését leíró modell. Azoknak a →lapoknak a minimális halmaza, amelyeket az →operatív memóriába kell tölteni ahhoz, hogy a program felesleges →laphibák nélkül, hatékonyan működjék. A helyesen megválasztott – feltétele a lapforgalom és a felhasználatlan memóriatér közötti dinamikus egyensúly. A – alapján javasolt →laphelyettesítési algoritmus tehát előírja, hogy egy programot csak akkor szabad végrehajtás céljából az operatív memóriába betölteni, ha –a számára elég hely áll rendelkezésre, ha viszont ezt a határt végrehajtás köz-

ben túllépték, a program minden pillanatnyilag aktív lapját dezaktiválni kell (ki kell menteni a →háttértárba).

munkarekesz (working storage) – olyan memóriarekesz v. -regiszter, amely a programban a közbenső részeredmények tárolására van fenntartva.

munkaütemező (job scheduler) – →kötegelt feldolgozásnál a →monitornak az a része, amely a munkafázisok és →munkák automatikus láncolását, a végrehajtásokhoz szükséges →erőforrások biztosítását, a végrehajtások elindítását végzi. Általában a rendszerlemezen található, és a →központi monitor kezdeményezésére kerül az operatív memóriába, minden egyes munkafázis befejezése után. A ~ legfontosabb feladata a →munkavezérlő nyelv parancsainak értelmezése és az igényelt funkciók elvégzése.

munkavége (end-of job) – →kötegelt feldolgozásnál a →munkavezérlő nyelv speciális parancsa (→vezérkártyája) egy-egy →munka lezárásának jelzésére. Feldolgozása egy sor tevékenységet indít el, közöttük a legfontosabb a munka számára kiutalt →erőforrások felszabadítása.

munkavezérlő nyelv (job control language) – egyes konkrét, →kötegelt feldolgozásban működő →operációs rendszerekre jellemző, speciális rendeltetésű →programozási nyelv (→parancsnyelv). Segítségével a felhasználó közölni tudja az operációs rendszerrel a különböző →munkák kezelésével kapcsolatos tennivalókat (pl. →fordítóprogramok v. más rendszerprogramok munkába állítását, →tárgyprogramok betöltését az operatív memóriába és azok végrehajtását, a végrehajtás számára szükséges szoftver, ill. hardver →erőforrások kijelölését stb.).

mutató (pointer) – adatstruktúrák fizikai tárolására alkalmazott, az egyes adatelemekhez csatolt segédadatmező. Rendszerint annak az adatelemnek a címét tartalmazza, amely az adott elem után (v. előtt) logikai sorrendben következik az adatstruktúrában.

A felhasznált cím természete szerint három típusa használatos. A →gépi című ~ gyors visszakeresést biztosít, amelyet elsősorban a közvetlenül címezhető adathordozókon szervezett láncolt típusú [adat]állományoknál, ill. az →operatív memóriában ábrázolt adatstruktúráknál használnak; hátránya a hangsúlyozott hardverfüggőség. A relatív című ~ lassúbb visszakeresést, de bizonyos gépfüggetlenséget biztosít. A szimbolikus című ~ még lassúbb visszakeresést biztosít; a logikailag következő (v. előző) adatelem azonosítóját használja címként; a legmagasabb gépfüggetlenséget nyújtja, de a valódi cím leképzése bonyolult algoritmust követelhet, ami a visszakeresést esetleg erősen lassítja.

mutató [adat]típus (pointer [data] type) – →erősen típusos nyelvekben a →dinamikus változók elérésére használható →[adat]típus. Értékkészletéhez tényleges memóriacímek tartoznak, megengedett műveletei az értékadás, ill. összehasonlítás (egyenlő, különböző). A ~ú változókon keresztül elérhető objektumok mindig ugyanahhoz az alaptípushoz kell hogy tartozzanak. Programozástechnikai szempontból igen jelentős, mert segítségével könnyen képezhetők összetett, rugalmas, sőt rekurzív adatstruktúrák is.

művelet (operation) – közkeletű értelmén kívül (mint eljárás, tevékenység), számítástechnikai vonatkozásban gyakran az →utasítás helyett is használják.

N

nagyfokú integráltság (large scale of integration) – a harmadik generációs →integrált áramkörökre jellemző technológia, amely egy →morzsán 100-nál több hagyományos logikai kapunak megfelelő funkcionális egységet képes előállítani.

navigálás (navigation) – számítástechnikai értelmezésben az →adatbázisokkal kapcsolatos művelet: a logikai és a fizikai adatszerkezet ismeretében meghatározott adatoknak ismert adatkapcsolatok alapján történő előkeresése. Ha az →adatkezelő nyelv →befogadó nyelvbe épül, akkor a programozó navigál (írja le a visszakeresés részletes műveletét), ha viszont önálló nyelv, akkor az →adatbázis-kezelő rendszer feladata a ~ elvégzése.

negáció (negation) – egyargumentumú →logikai függvény, amelynek értéke mindig az argumentum ellentettje. Igazságtáblázata:

A	$\bar{A}$
0	1
1	0

negatív visszaigazolás (negative acknowledge) – adatátviteli műveletekben a hibaellenőrzés olyan módszere, amely bármilyen, felismerhető hibával vett üzenetet megismételtet. A visszatérő jel, amely a hibát jelzi, a NAK (a CCITT 5-ös számú ábécéjében). Az →automatikus ismétléskéréssel dolgozó rendszerek többségében az adóállomás a helyes vételről →visszaigazolást (ACK) igényel, csak kevés rendszer tekinti a ~ hiányát hibátlan átvitelnek.

négycímű számítógép (four-address computer) – olyan →digitális számítógép, amelynek utasításaiban 4 →cím szerepel. Az első kettő rendszerint a műveletben szereplő két operandus címe, a harmadik az eredmény tárolási címe, míg a negyedik a soron következő végrehajtandó utasítás címe.

negyedik generációs számítógép (fourth-generation computer) – kb. 1972-től napjainkig gyártott →integrált áramkörös →digitális számítógépek elnevezése (kapcsolási sebességük nanosec alatt). A →harmadik generációs számítógéppel szemben inkább mennyiségi, mint minőségi változásokat hozott. →Architektúrája még mindig a klasszikus →Neumann-gép elvén alapszik, csak működési paraméterei különböztetik meg az előző generációtól: az igen nagy műveleti sebesség, a mind kisebb kiterjedésű, kompakt felépítés, a mind nagyobb →operatív memóriakapacitás, az egyre nagyobb tárolási sűrűség, az egyre csökkenő ár és növekvő megbízhatóság jellemzik. Szervezési és programozástechnikai szempontból az →elosztott vezérlés, az →elosztott feldolgozás, a →számítógép-hálózatok megjelenése jelent újat. Külön kategóriát, de egyáltalán nem lebecsülendő teljesítményt hordoznak az utóbbi 20 évben megjelent s látványos karriert befutott →mikroszámítógépek, ill. →személyi számítógépek.

nemdeterminisztikus automata (nondeterministic automaton) – olyan →automata, amelynek következő állapota nem egyértelműen meghatározott a pillanatnyi állapot és a bemeneti szimbólum által, vagyis az automatánál felhasznált formális definíció alapján:

$$f(s,a)=\{q_1,q_2,\ldots,q_n\} \in S, \text{ ahol } a \in A,\ s \in S$$

Ez azt jelenti, hogy a pillanatnyi állapot és a bemenő szimbólum függvényében meg kell adni mindazokat az állapotokat, amelyeket a ~ felvehet.

nemdeterminisztikus program (non-deterministic program) – olyan →program, amelynek a végrehajtás alatti, időben bekövetkező állapotváltozásai nem határozhatók meg előre, vagyis nem létezik egy olyan leképzési függvény, amely a bemenő adatok alapján egyértelműen meghatározná a kimenő adatokat. Ilyen programokat állít elő a →logikai programozás (→Prolog).

nemlineáris programozás (non-linear programming) – a →lineáris programozással szemben minden olyan feladat megoldása, amelyben egy nemlineáris célfüggvény szélső értékeit (maximumát v. minimumát) kell meghatározni valamilyen lineáris egyenlőtlenség-rendszer nemnegatív megoldásai mellett (lineáris feltételű ~i feladat) v. nemlineáris egyenlőtlenségrendszer megoldásai mellett (nemlineáris feltételű ~i feladat). Az első osztályba olyan feladatok tartoznak, amelyek megoldhatók →szimplex módszerrel.

nemnumerikus literál (non-numerical literal) – idézőjelekkel határolt karaktersorozat,

→literál, amely – az idézőjelektől eltekintve – a számítógéprendszer által felismerhető tetszőleges karaktereket tartalmazhat.

nemprocedurális nyelv (non-procedural language) – →deklaratív nyelv

nemszabványos címke (non-standard label) – 1. →Mágnesszalagon tárolt →[adat]állományokra jellemző, egyes →operációs rendszerekben megengedett olyan →címke, amelyet az operációs rendszer elolvas (ill. kiír) ugyan, de figyelmen kívül hagy, így feldolgozásáról a felhasználónak kell gondoskodnia. – 2. Nem szabványos szektorfej a merev- v. hajlékonylemezen, a másolásvédelemben alkalmazzák.

nemterminális szimbólum (non-terminal symbol) – a →formális nyelvek elméletében nyelvtani osztályokat jellemző kifejezés. A ~ nem jelenik meg mint olyan a definiálandó nyelv →mondataiban, csak a →mondatformák generálásánál (leképzésénél) egy-egy fázis (lépés) azonosítására szolgál. ~ pl. a legtöbb programozási nyelvben definiált →azonosító, →címke, →kulcsszó stb.

A ~ok halmazának van egy kitüntetett eleme, a →kezdőszimbólum, amelyből egy nyelvet generáló nyelvtan →helyettesítési szabályai alapján eljutnak az illető nyelvhez tartozó bármelyik mondatformához. A természetes nyelvek viszonylatában a ~ok olyan nyelvtani kategóriáknak felelnek meg, mint az alany, az állítmány, a tárgy stb.

nemvégrehajtható utasítás (non-executable statement) – magas szintű →programozási nyelvekre jellemző utasítás, amelynek nem marad konkrét nyoma a lefordított →tárgyprogramban, viszont kihatással van a program generálására és végrehajtására. A leggyakrabban használt ~ok a →deklaratív utasítás, a memória kiosztására vonatkozó utasítás, az →[adat]típusokat deklaráló utasítás, az adatformátumokat, nyomtatási sablonokat leíró utasítások stb.

Neumann-gép (von Neumann machine) – a napjainkig ismeretes →digitális számítógépek alapjául szolgáló, az ún. Neumann-elvet megvalósító elméleti modell, amelyet szerzője, Neumann János (Arthur W. Burks és Hermann H. Goldstein közreműködésével) 1947-ben közölt munkájában hozott nyilvánosságra. A Neumann-elv központi meghatározásai az adatokkal együtt tárolt vezérlőprogram, az utasításvezérlés és a szekvenciális végrehajtás.

A ~et megvalósító →architektúra alapvető funkcionális jellemzői: a →központi vezérlőegységet és az →operatív memóriát magában foglaló központi egység; az egyszintű fizikai →címtartomány; a centralizált szekvenciális vezérlés (a tárolt program alapján); az alacsony szintű →gépi nyelv.

név szerinti hivatkozás (call by name) – az →eljárások hívásánál (aktivizálásánál) használatos paraméterátadási módszer. Lényege, hogy az →aktuális paramétereket a hívott eljárás törzsében a megfelelő →formális paraméterek helyébe szövegszerűen kell átmásolni. A ~ a →cím szerinti hivatkozástól elvileg abban különbözik, hogy az aktuális paraméter címét mindannyiszor ki kell számítani, valahányszor a megfelelő formális paraméter a hívott eljárásban előfordul.

névtartomány (name space) – mindazoknak a szimbolikus neveknek, azonosítóknak halmaza, amelyeket a programban (programegységben) tárolt információk hivatkozására felhasználnak (→címtartomány, →érvényességi tartomány).

normalizálás (normalization) – →lebegőpontos számábrázolásban kódolt számok ún. normalizált alakra való átalakítása. Az

$$m \cdot a^c$$

számokról akkor mondják, hogy normalizált formájúak, amikor a →mantisszára érvényes a következő kifejezés:

$$1/a < m < a$$

vagyis a mantissza a választott ábrázolási mód figyelembevételével a lehető legnagyobb szám. Ez azt jelenti, hogy m a fenntartott biteken még ábrázolható, de az $a \cdot m$ már nem.

Előfordulhat, hogy a szám ~a nem hajtható végre, mert a →karakterisztikára olyan nagy negatív szám adódna, amely már a számára fenntartott biteken és a választott kód mellett nem ábrázolható. Az ilyen számokat rendszerint gépi nulláknak tekintik. A lebegőpontos →aritmetikai egységgel számítógépekben az aritmetikai műveletek eredményét rendszerint automatikusan ~nak vetik alá, a

felhasználó mindig normalizált eredményekre számíthat.

növekményes címzés (autoincremental addressing) – elsősorban kisszámítógépeknél használatos címzési módszer. Lényege az, hogy a →gépi utasítás címrésze annak a regiszternek a címét tartalmazza, amelynek tartalma a tulajdonképpeni operandus címe. Az utasítás végrehajtása után az illető regiszter tartalma automatikusan egy szóhosszal (rendszerint 2 →bájttal) nő (v. csökken), előkészítve ezáltal a következő feldolgozandó operandus címét. A 8-32 regiszter címzésére 3-5 bit elégséges, és így a gépi utasítás nem foglal el többet 2 bájtnál. Ugyanakkor kényelmes módszer tömbök, táblázatok lépésenkénti feldolgozására.

numerikus adatok (numerical data) – a környező valós világ főleg mennyiségi összefüggéseit számokkal, számjegyekkel kifejező →információinak számítógépes ábrázolása.

numerikus analízis (numerical analysis) – a matematika egyik legrégebben művelt ága, amely a matematikai v. más tudományok területén megfogalmazott feladatok esetenként pontos, de legtöbbször megközelítő megoldásával foglalkozik, numerikus módszerek alapján. Lényege az, hogy az adott feladat megoldására →algoritmust ad, amely véges számú elemi művelet után az eredmény megközelítéséhez vezet.

A ~ az elektronikus számítógépek elterjedésével újabb jelentőséget nyert, ugyanis sok módszer, képlet csak a számítógépek által nyújtott előnyök révén válik gyakorlatilag is használhatóvá.

numerikus kód (numerical code) – olyan →kód, amelynek →ábécéjét csak számjegyekkel v. számokkal ábrázoljuk. A számítástechnikában használatos legtöbb kód tkp. ~, ugyanis végső esetben mindegyik →bináris ábrázolásra redukálható. Ide tartoznak a →BCD, →EBCDIC, →ASCII stb., de tágabb értelmezésben a →vonalkód is.

numerikus konstans (numerical constant) – olyan →konstans, amely a számjegyekből, előjelekből és tizedespontból álló karakterhalmaz elemeiből építhető fel, s képviselhet egész, valós, esetleg komplex számot. Az előjel, ha előfordul, csak a balról számított első karakter lehet. Egyes programozási nyelvek megengedik a lebegőpontos ~ használatát is, ez egy →mantisszából és egy exponensből épül fel (az exponenst rendszerint „e" betű vezeti be).

numerikus literál (numerical literal) – →numerikus konstans

numerikus módszer (numerical method) – bármilyen komplex feladatnak csak a négy alapművelet segítségével történő megoldása, közelítő számítása. A kiszámítandó mennyiséget közelítő számokkal, közelítő függvényekkel approximálják. A függvényekkel való megközelítéseknél sokszor van szükség arra, hogy a folytonos változásokat leíró folytonos függvények értékeit csak diszkrét helyeken vizsgálják (időben és amplitúdóban).

numerikus stabilitás (numerical stability) – a matematikai →modellnek adott numerikus módszerrel kapcsolatos érzékenységét jellemzi, vagyis azt, hogy a modell bizonyos paramétereinek változásait a numerikus módszer hogyan viszi át az eredménybe. A paraméterek változásait olyan külső behatások érhetik, mint pl. a számítógépben történő ábrázolás, amely legtöbbször a valódi érték megközelítése. A ~ feltételezi, hogy a paraméterek ilyen változásai mellett a nyert eredmény elfogadható határokon belül közelíti meg a valódi eredményt.

numerikus számítógép (numerical computer) – →digitális számítógép

numerikus vezérlés (numerical control) – különféle berendezések (elsősorban szerszámgépek) olyan vezérlése, amelynél az előre ismert műveletek előre ismert sorrendben történő elvégzését numerikusan kódolt vezérlőadatokkal biztosítják. A vezérelt berendezés vezérlőegysége felismeri ezeket a kódokat, és nekik megfelelően vezérli a berendezés alegységeit. A vezérlőadatok →lyukszalagon, →lyukkártyán, →mágnesszalagon v. →mágneslemezen lehetnek rögzítve.

NY

nyelv (language) – az információközlés alapvető eszköze. Három meghatározó jellegű összetevője az információ kódolási formáit és azoknak mondatokba történő szerkesztési szabályait magában foglaló →szintaxis, a ~ mondataihoz rendelt értelmezés, a →szemantika, valamint a ~ felhasználását meghatározó szabályokat összefoglaló pragmatika.

Kialakulási körülményeik alapján a ~ek két nagy csoportra oszthatók: természetes és →mesterséges nyelvek. Számítástechnikai szempontból különleges jelentőséggel rendelkeznek a →programozási nyelvek.

nyelvi processzor (language processor) – →fordítóprogram, →értelmezőprogram v. más mechanizmus, amely forrásnyelvű programot fogad el bemeneti adatként, és azt a nyelv definíciójában leírtak szerint értelmezni képes.

nyelvtan (grammar) – valamely (természetes v. →mesterséges) nyelv mondatszerkezeteit meghatározó szabályok összessége. A mondatszerkezetek leírását célszerű úgy megadni, hogy a mondatokat részekre bontják és az egyes részek felépítését külön határozzák meg. Ezeket a részeket olyan szimbólumokkal azonosítják, amelyek nem tartoznak a definiálandó nyelv szógyűjteményébe.

A ~ leírásához tehát két diszjunkt halmazra van szükség: a →terminális szimbólumok halmazára, amely tkp. az illető nyelv ábécéje, és a →nemterminális szimbólumok halmazára, amely a mondatszerkezetek egy-egy részének azonosítására szolgáló szimbólumokból áll; ezek között van egy kitüntetett szimbólum, amely feltétlenül megjelenik önmagában legalább egy →helyettesítési szabály bal oldalán, ez a →kezdőszimbólum.

Ezekkel az elemekkel megadható a ~ formális meghatározása: a ~ egy (T,N,S,P) négyes, ahol T a terminális szimbólumok halmaza, N a nemterminális szimbólumok halmaza, S a kezdőszimbólum (v. mondatszimbólum), P pedig a helyettesítési szabályok véges halmaza.

A ~ok osztályozása N. Chomskytól származik, aki a helyettesítési szabályokra tett megszorítások alapján a ~ok 4 osztályát különböztette meg (→Chomsky- hierarchia). Adott ~ által generált mondatok halmaza az illető nyelv.

A természetes nyelvek feldolgozásához, ill. a modern fordítóprogramok előállításához ennél fejlettebb nyelvtanokat is használnak.

nyelvtani konstans (grammatical constant) – →terminális szimbólum

nyelvtani változó (grammatical variable) – →nemterminális szimbólum

nyílbillentyű (arrow key) – a →billentyűzet azon billentyűje, amelynek segítségével a →helyőr képernyőn elfoglalt helyét lehet változtatni. Rendszerint négy ilyen speciális billentyű létezik a helyőrnek lefelé v. felfelé, ill. jobbra v. balra mozgatására.

nyílt hatásláncú vezérlés (open-loop control) – a →folyamatvezérlés olyan megvalósítása, amelynél az ellenőrzés és a tényleges vezérlés közé emberi beavatkozás épül, esetleg az egyik fázis teljesen hiányzik. A →folyamatvezérlő számítógép a folyamat állapotváltozásait közvetlen mérések útján követi, s e mérési adatok feldolgozása révén segíti a döntéshozatalban az emberi operátort.

nyílt rendszerű összekapcsolás (open system interconnection) – különböző →protokollokon alapuló →adatállomásoknak →adatátviteli hálózatok segítségével történő összekapcsolását mint egységes logikai hierarchiát vizsgáló elméleti modell. Az összekapcsolási szabványokat az ISO dolgozta ki, ennek eredménye a →hétrétegES referenciamodell.

nyílt szubrutin (open/direct-insert subroutine) – olyan rövidebb-hosszabb utasítássorozat, amely kisebb-nagyobb változtatásokkal többszörösen beépül egy programba. Rövidebb utasítássorozat ismétlődésekor nem célszerű megterhelni azok végrehajtását a →zárt szubrutin aktivizálásához szükséges újabb utasításokkal (paraméterátvitel, munkaregiszterek mentése, visszatérési cím rögzítése, vezérlésátadás). Jellegzetes ~ a →makroutasítás.

nyilvános adathálózat (public data network) – nyilvános felhasználók számára fenntartott, általában →csomagkapcsolásos üzemmódban dolgozó →adatátviteli hálózat. Üzemeltetője rendszerint a specializált kormányzati szerv (PTT), vezérlése valamely fej-

lett (X.25–X.400) →protokollon és a vele kapcsolatos CCITT szabványokon alapul.

nyilvános vonalkapcsolásos telefonhálózat (public switched telephone network) – a hagyományos hangfrekvenciás telefonhálózat hivatalos megnevezése a CCITT szabványokban (→vonalkapcsolásos hálózat).

nyitószabály (initiating routine) – →döntési táblázatoknál az a teendő, amely a feltételektől független és csak egyszer, a végrehajtás elején kell vele foglalkozni.

nyitott gépterem (open shop) – →kötegelt feldolgozással üzemelő számítóközpont szervezési módja. Jelentése az, hogy a végrehajtandó program tulajdonosa (a számítóközpont bérlője) hozzáférhet a számítógéphez, ha szükséges, beavatkozhat a végrehajtás menetébe (szemben a →zárt gépterem szervezési móddal).

nyolcas számábrázolás (octal representation) – →oktális számábrázolás

nyomkövetés (trace) – →programok belövési (tesztelési) fázisában a logikai hibák felderítésére használatos módszer, amelynek segítségével információkat lehet gyűjteni a vizsgált program végrehajtása alatt keletkező átmeneti eredményekről, egyes változók állapotváltozásairól, a bejárt logikai programágakról. Kötegelt feldolgozásnál az adatok a végrehajtás során az ún. nyomkövetési tárba kerülnek, időszakonként v. leállás után onnan – már az emberi észleléshez és kiértékelhetőséghez alkalmas ütemben és formában – kiolvashatók és értelmezhetők. A ~ módszere elsősorban a →kötegelt feldolgozásra jellemző, →párbeszédes üzemmódban kényelmesebb s főleg hatékonyabb módszerek állnak rendelkezésre (→lépésenkénti végrehajtás, →ellenőrzött végrehajtás). Egyes →fordítóprogramok is lehetővé teszik a ~t v. megfelelő eszközökkel, v. a →feltételes fordítás megvalósításával.

nyomtatási mező (printing output area) – valamely →programon belül olyan adatzóna, amely rendszerint egy kinyomtatandó sornak (de esetleg egyetlen adatelemnek) felel meg, ahol az összegyűjtött adatokat a nyomtatási sablonnak megfelelően előkészítik. A ~ben rendszerint a közvetlenül nyomtatható →hexadecimális, →ASCII v. →EBCDIC kódolási formát használják.

nyomtatási sablon (printing picture) – numerikus adatok nyomtatási szerkezetét meghatározó →nemvégrehajtható utasítás (pl. FORMAT a →FORTRAN-ban, PICTURE a →COBOL-ban v. →PL/1-ben stb.). Rendszerint a nyomtatandó tizedespozíciók számára, a vezető nullák elnyomására, a gazdasági-ügyviteli feldolgozás számára szükséges speciális szimbólumok beékelésére stb. vonatkozik.

nyomtató (printer) – a számítógéprendszer által szolgáltatott információkat a felhasználó által közvetlenül értelmezhető és felhasználható formában papírra rögzítő →kimeneti egység, amely alfanumerikus formában, papírszalagon v. egyenként adagolt íveken teszi hozzáférhetővé az információkat. →Kötegelt feldolgozásnál a standard kimeneti egység rendszerint ~. Működési elvük szerint a ~k lehetnek soros típusúak (ezeknél egy írófej mozgatásával valósul meg egy sor nyomtatása) és párhuzamos működésűek, vagyis egyszerre egész sor (sornyomtató) v. lap (lapnyomtató) kinyomtatására alkalmasak.

O

objektum alapú módszer (object-based method) – →ismeretábrázolásban (is) használható programozási módszer, amely objektumoknak nevezett elemeket alkalmaz. Az objektumok olyan önálló egységek, amelyek egyszerre tartalmazzák az információt és a kezelésére szolgáló algoritmusokat. Ehhez az információhoz nem lehet hozzáférni, csak az egyedi kezelőalgoritmusokon keresztül.

objektumorientált nyelv (object-oriented language) – →objektum alapú módszert támogató magas szintű →programozási nyelv. Alapvető jellegzetessége, hogy olyan típusú →információrejtést tesz lehetővé, amelynél a felhasználó →absztrakt [adat]típusokat definiálhat a hozzá tartozó kezelő algoritmusokkal együtt (együttes nevén objektumokat). Ezeknek konkrét megvalósítását utólag független módon lehet változtatni, a felhasználói program módosítása nélkül. Egyik legismertebb ~ a SMALLTALK, de ebbe a családba sorolhatók a Borland cég TURBO C++, ill. 6.0-s termékei is.

off-line – az angol szakirodalomból átvett kifejezés, a számítógéprendszerhez tartozó olyan berendezés, amely nem csatlakozik közvetlenül hozzá, vagyis nincs a →központi vezérlőegység felügyelete alatt. Ilyen alapon ~ berendezés a pillanatnyilag kikapcsolt állapotban lévő →periferikus mű is.

oktális számábrázolás (octal representation) – a számoknak 8 hatványai szerint rendezett, vagyis oktális számrendszerben felírt ábrázolása. Az együtthatók jelkészlete {0,1,2,3,4,5,6,7}. Minthogy $8=2^3$, három bináris számjegy foglalható össze egyetlen oktális számjeggyé.

oktatógép (teaching machine) – olyan gép, berendezés, amellyel oktatási feladatok végezhetők. A modern ~ rendszerint számítógépet is magában foglaló, →párbeszédes üzemmódban működő berendezés. Az adott területre jellemző kérdésekre, feladatokra a tanuló →terminálon keresztül adja meg a válaszokat, eredményeket, az ~ pedig kiértékeli ezeket a válaszokat, esetleg megadja a helyes eredményeket stb.

oktett (octet) – nyolc egymás melletti →bit együttesen kezelt csoportja. A számítógépes →architektúrák túlnyomó többségében a →bájttal azonos fogalom, bár ismertek olyan gépek is, ahol a bájt 7,9, esetleg 12 bit csoportját jelképezi.

olvasás (reading) – számítástechnikai vonatkozásban az alábbi három művelet egyike: 1) beolvasás: az információnak valamely külső →adathordozóról (→lyukkártyáról, →mágneslemezről, →mágnesszalagról stb.) az →operatív memóriába történő átvitele; 2) kiolvasás: az operatív memóriában tárolt információ átvitele a →központi vezérlőegység regisztereibe, további feldolgozás céljából; 3) szöveg beolvasása optikai karakterolvasó és -felismerő rendszerrel.

olvasófej (read head) – olyan (mágneses, optikai stb.) szerkezet, amely →adathordozókon rögzített adatok visszakeresésére, visszanyerésére alkalmas. Így pl. a mágneses ~ a mozgást végző mágneses közeg felett van elhelyezve, és annak a rögzített információra jellemző mágneses állapotát érzékeli.

on-line – az angol szakirodalomból átvett kifejezés olyan berendezés jelölésére, amely úgy kapcsolódik a számítógéphez, hogy a vezérlést befolyásoló emberi beavatkozásra nincs szükség. Az ~ berendezés v. közvetlenül a →központi vezérlőegység felügyelete alatt áll, v. speciális →adatcsatorna vezérli.

opcionális szó (optional word) – olyan →lefoglalt szó, amelynek használata a megfelelő nyelvi szerkezetben megengedett, de nem kötelező, jelenléte csak a program olvashatóságának könnyítésére szolgál.

operációkutatás (operations research) – bonyolult ipari, kereskedelmi, gazdasági, társadalmi rendszerek felépítését, szervezését, vezetését, irányítását stb. matematikai eszközökkel tanulmányozó, kiértékelő, ill. modelláló tudományág. Alapvető eszközei a valószínűség-számítás, a matematikai statisztika, a →játékelmélet, a →matematikai programozás, a →várakozási stratégiák stb. Legfontosabb célja a komplex rendszerekben megfigyelhető műveletek mérése, modellálása és szimulálása, majd ezeken keresztül a megadott feltételek közötti optimális döntések meghatározása. Az ~ általános módszere olyan tudományos modellek megalkotása, amelyekben a determinisztikus elemek mellett véletlen, sztochasztikus jellegű tényezők

is szerepet játszanak. Ezekkel a modellekkel előre megállapíthatók a döntés stratégiái v. irányítási alternatívái.

operációs rendszer (operating system) – számítógépek teljesítményének növelésére, az emberi beavatkozás minimálisra csökkentése érdekében előállított programcsomag, amely az összes →hardver és →szoftver →erőforrás kezelését és irányítását úgy hangolja össze a különböző felhasználók részére, hogy minél kevesebbet álljanak kihasználatlanul. Kielégítően oldja meg azokat a konfliktushelyzeteket, amelyek akkor állhatnak elő, amikor több felhasználó igényli egyszerre ugyanazt az erőforrást. Bár a gyakorlati megvalósítások számtalan módozata ismeretes, az ~nek a következő fő feladatokat kell megoldania:

1) A →központi vezérlőegység kezelése: az →operatív memóriában egyszerre jelen lévő programok, feladatok nyilvántartása, azok állapotának kimerítő ismerete annak céljából, hogy mindig az a program jusson hozzá a központi vezérlőegységhez, amely azt ténylegesen fel is tudja használni; a →prioritási rendszeren alapuló →multiprogramozás támogatása.

2) A →periferikus művek kezelése: ezek kiosztása és felszabadítása, az adatcserékkel kapcsolatos műveletek kezdeményezése, követése és befejezése; a periferikus egységeket vezérlő →adatcsatornák állandó figyelése, állapotváltozásaik nyomon követése a minél magasabb fokú kihasználtság érdekében; a →puffertechnika megvalósítása.

3) Az operatív memória kezelése: a fix v. változó számú és hosszúságú →partíciók kiosztása és felszabadítása, a memóriavédelemmel kapcsolatos igények kielégítése.

4) Az →[adat]állományokkal kapcsolatos műveletek hatékony elvégzése.

5) A felhasználóval (v. gépkezelővel) folytatott párbeszéd, információcsere lebonyolítása.

6) A rutinfeladatok elvégzésére hivatott →rendszerprogramok kiutalása és az azok által igényelt erőforrások biztosítása (→fordítóprogramok, →értelmezőprogramok, →kapcsolóprogramok, →betöltőprogramok, →[adat]-állomány-kezelő rendszer, →rendezőprogramok, →könyvtárkezelő programok stb.).

7) A számítógéprendszer üzemeltetésével kapcsolatos nyilvántartás elvégzése.

A felhasználó és az ~ közötti dialógus jellegzetes formális nyelv (→parancsnyelv) segítségével valósul meg, amellyel a felhasználó leírhatja a megoldandó munka jellemzőit, az igényelt erőforrások számát és természetét.

operandus (operand) – valamely művelet tárgyát képező mennyiség, szigorúbb számítástechnikai értelmezésben a →gépi utasítás →címrészének értelmezése révén nyert adat. Ilyen megközelítésben az ~ lehet valódi, amikor a címértelmezés révén az →operatív memória egy (v. több) rekeszének a tartalma az ~, de lehet olyan információ is, amely az utasításra vonatkozó jellemzőket rögzít (pl. a →léptetésnél, ahol a művelet típusára, módjára, valamint a lépések számára vonatkozik). →Digitális számítógépek vonatkozásában az ~ lehet egy szám (→numerikus literál), egy karakter, egy jelsorozat, egy táblázat stb.

operanduslehívás (operand fetch) – az utasítás-végrehajtás egyik fázisa. A →központi vezérlőegység által végrehajtott művelet, amelynek révén a →gépi utasítás címrészében található operandus(ok) címét felismeri, és az ott található operandus(oka)t az →operatív memóriából a saját belső regisztereibe átviszi.

operatív címtartomány (operative address space) – mindazok az →operatív memóriához tartozó címezhető lokációk, amelyekben információ tárolható (→címtartomány, →névtartomány, →logikai címtartomány).

operatív memória (operative memory) – a →központi vezérlőegység által közvetlenül címezhető, elérhető tárolóegység, amely a végrehajtás alatt álló program(ok) utasításait és adatait tartalmazza. Olyan →véletlen hozzáférésű memória, amelynek elérési ideje általában 1s alatt van. A →digitális számítógép architektúrájának alapvető része a →központi egység, amely a központi vezérlőegység és az ~ együttese. (Az ~ szinonimái: belső memória, elsődleges memória, központi memória.)

operátor (operator) – 1. Valamely →programozási nyelvet generáló →nyelvtan →terminális szimbóluma. Az ~ az operandusokkal elvégzendő műveletre vonatkozó műveleti jel. A műveletben részt vevő operandusok számának megfelelően az ~ lehet egyoperandu-

sú (unáris), kétoperandusú (bináris) stb. Az operandusok, ill. az eredmény típusának megfelelően az ~ lehet aritmetikai (aritmetikai operandusok és aritmetikai eredmény), relációs (aritmetikai operandusok és logikai eredmény), logikai (logikai operandusok és logikai eredmény), jelsorozatokkal műveletet végző ~ stb. – 2. Számítógép-kezelő.

optikai karakterfelismerés (optical character recognition) – optikai eszközökkel és módszerekkel végzett →karakterfelismerés.

optikai karakterolvasó (optical character reader) – olyan berendezés, amely a géppel v. kézzel írott szövegeket a számítógép által közvetlenül feldolgozható formára alakítja.

optikai lemez (optical disk) – →kompaktlemez

optimalizálás (optimization) – általános értelmezésben valamely feladat olyan megoldása, amely legjobban megfelel a környezet által támasztott feltételeknek. Számítástechnikai vonatkozásban legtöbbször két alapvető kritérium áll az ~ központjában: a →végrehajtási idő minimalizálása v. a programtermék által igényelt →operatív memória nagysága (→programoptimalizálás).

ortogonális adatstruktúra (orthogonal data structure) – a →lineáris adatstruktúra egyik általánosítása, a teljesen v. hézagosan kitöltött n dimenziós →tömbökre vonatkozik ($n > 2$). A kétdimenziós tömböket rendszerint szekvenciálisan ábrázolják az →operatív memóriában. Az ábrázolás sorfolytonos, amikor az első →index szerint helyezik el sorrendben az elemeket, és oszlopfolytonos, amikor a második index a mérvadó. Sorfolytonos ábrázolásnál, háromdimenziós tömböt véve alapul, egységnyi elemhosszúságot feltételezve, az x(i,j,k) adatelem címe a következő összefüggés alapján határozható meg:

$$loc(x(i,j,k)) = loc(x(1,1,1)) + n_2 \cdot n_3 \cdot (i-1) + n_3 \cdot (j-1) + (k-1)$$

feltéve, hogy az indexek értéke 1-től indul.

OS/VS (Operating System for Virtual Storage) – az IBM nagyszámítógépeire kifejlesztett →operációs rendszer, amelynek alapvető jellemzője a →virtuális memória kezelése.

oszcilláló összeválogatás (oscillate merging) – az →összeválogatásnak a →külső rendezés megvalósítására használt módszere. Feltételezve, hogy n a felhasznált →periferikus egységek száma, az ~ lényege a következő: valamilyen belső rendezési módszerrel monotóniákat állítanak elő, s az első lépésben n–1 perifériára írnak ki egy-egy rendezett részhalmazt. A következő lépésben a rendezett részhalmazokból egy-egy elemet beolvasva és a rendezési elv szempontjából sorba állítva, az n-edik perifériára kiírják őket. Ezt a lépést még n–2-szer megismétlik, mindig más egységet hagyva üresen. Ilyen módon n–1 egységen kapnak egy-egy rendezett részhalmazt, amelyek mindegyike n–1-szerese a belső rendezéssel előállítható halmazoknak. Ezeket összeválogatják egy részhalmazba az éppen üres egységen, és ismét kezdik az eljárást elölről. Amikor azonos nagyságú részhalmazok vannak n–1 egységen, mindig összeválogatják őket az n-edikre. A szétosztást addig folytatják, amíg egészében rendezett halmazt nem nyernek.

osztott erőforrás (shared resource) – olyan →hardver v. →szoftver →erőforrás, amely látszólag v. ténylegesen egyszerre több felhasználó rendelkezésére áll. Jellegzetes ~ a →központi vezérlőegység ideje az →időosztásos rendszerekben, az →operatív memória a →lapkezeléses rendszerekben, az →operációs rendszer nemrezidens részét tartalmazó mágneslemezegység stb. Szoftver ~ok pl. a →felügyelőprogram szubrutinjai stb. →Adatátviteli hálózatok vonatkozásában ~ a →kommunikációs csatorna. Az előfizetők közötti információcsere megvalósítására a következő kapcsolási módszerek használatosak: →vonalkapcsolás, →üzenetkapcsolás, →csomagkapcsolás.

osztott vezérlés (shared control) – az →adatbázis-kezelő rendszer üzemmódja, amellyel lehetővé teszi, hogy többfelhasználós környezetben az adatelemekhez egyidejűleg több felhasználói program is hozzáférjen.

Ö

önálló nyelv (self-contained language) – az →adatbázisokkal kapcsolatos olyan →programozási nyelv (→adatleíró nyelv v. →adatkezelő nyelv), amely céljának megfelelő saját →szintaxissal és →szemantikával rendelkezik, s ezért saját →fordítóprogramra is szüksége van. Az adatleírásra használt egyik ~ pl. az IBM DL/I, amely az adatbázis logikai (→adatbázisséma, →adatbázis-alséma) és fizikai ábrázolásának meghatározására egyaránt alkalmas.

önelőkészítő program (self-initializing program) – olyan →program, amely minden egyes végrehajtás előtt maga gondoskodik lokális változóinak kezdőértékadásáról (így újrabetöltés nélkül többször is végrehajtható).

önkomplementáló kód (autocomplementing code) – decimális számok kódolásának olyan fajtája, amelynek alapvető tulajdonsága az, hogy adott szám kódjának minden egyes bináris jegyét komplementálva, az eredeti szám kilences komplementumát nyerik.

öntöltő program (self-loading program) – olyan →program, amely az →operációs rendszer (ill. annak speciális komponense, a →betöltőprogram) közreműködése nélkül is betölthető és végrehajtható. Jellegzetes ~ pl. a →hardveregységeket ellenőrző tesztprogram.

öröklött attribútum (inherited attribute) – a D. Knuth által javasolt →attribútumnyelvtanban használt fogalom. Az adott nyelvtan →nemterminális szimbólumához kapcsolt olyan attribútumra vonatkozik, amelyet az illető nyelvtani változó felülről kap.

öröklött hiba (a priori/inherited error) – valamely számításhoz adott kiinduló adatok minden olyan hibája, amely mérésekből, megelőző számításokból, adatbeviteli v. ábrázolási hibákból származik. Az ~ a számítás egész menetén végigvonul, és ilyenformán az eredményt is erősen módosíthatja. Rendszerint →hibaképlet segítségével fejezik ki, hogy az ~ mennyire befolyásolhatja az adott eljárást.

összehasonlítás (comparison) – két operandus értékének egyenlőségét v. különbözőségét meghatározó művelet. Lehet alfanumerikus (jelsorozatok azonosságának megállapítására) v. numerikus (aritmetikai értékek egyenlőségének kimutatására). Ez utóbbi végezhető az előjelek figyelembevételével (algebrai ~) v. anélkül (aritmetikai ~).

összeköttetés (link) – 1. →Adatátviteli hálózatokban az a kommunikációs vonal, amely két →csomópontot v. kapcsolóközpontot összeköt. – 2. Elvi kapcsolás a →csomagkapcsolásos hálózatban lévő két fél között, annak jelzéseként, hogy egymással kommunikációs kapcsolatban lehetnek.

összeláncolás (concatenation) – az a művelet, amelynek során két tetszőleges (x és y) →karakterlánc (v. vektor) egymás után történő elhelyezésével előállítódik a karakterláncok (vektorok) xy rendezett sorozata. Az ~ asszociatív művelet, vagyis x és yz ~a azonos xy és z ~ával. Kiterjeszthető úgy, hogy jelsorozat helyett egyetlen jelre is alkalmazható legyen.

összetett feltétel (combined condition) – a →COBOL programozási nyelvben (többek között) használatos olyan szintaktikai kategória, amely →ÉS és/vagy →VAGY logikai operátorokkal összekapcsolt →egyszerű feltételekből épül fel.

összeválogatás (merging) – két v. több rendezett →logikai [adat]állománynak egyetlen rendezett [adat]állományba történő egyesítése. A kiinduló [adat]állományok rendezési kritériumai rendszerint meghatározzák az előállítandó [adat]állomány rendezési formáját is. Az ~ általában kiegészítő művelete a →külső rendezésnek, ill. az olyan rendezésnek, ahol a rendezendő adatmennyiség olyan nagy, hogy lehetetlen az →operatív memóriában megvalósítható →belső rendezés. Ebben az esetben valamilyen belső rendezési módszerrel rendezik az operatív memóriába betölthető részhalmazokat, majd kiírják azokat valamilyen mágneses adathordozóra (mágneslemezre v. mágnesszalagra). Attól függően, hogy miként történik a rendezett részhalmazok (monotóniák) ~a, van →egyenletes összeválogatás, Äoszcilláló összeválogatás és Ätöbbfázisú összeválogatás.

összeválogatásos rendezés (merge sorting) – a →rendezésnek az →operatív memóriában megvalósítható módszere. Lényege: a teljes rendezendő halmazt részhalma-

zokra tagolják és ezeket külön-külön rendezik valamilyen módszerrel. Az így nyert rendezett részhalmazokat párosával nagyobb halmazokká egyesítik összeválogatással, míg az utolsó két részhalmaz egyesítése a teljes rendezett adathalmazt eredményezi.

összevonó szintaktikai elemzés (reductive syntactical analysis) – →alulról felfelé haladó szintaktikai elemzés

P

palást (cylinder) – →cilinder

parallel algoritmus (parallel algorithm) – olyan →algoritmus, amelyben bizonyos résztevékenységek megfelelő felügyelet mellett egyidejűleg (párhuzamosan) végrehajthatók. A →multiprocesszoros rendszerek megjelenésével a ~ok olyan feladatok számítógépes megvalósítását is lehetővé teszik, amelyekben bizonyos részfeladatok időben átfedik egymást.

parallel (működésű) számítógép (parallel computer) – 1. →Digitális számítógép, amelynek →központi vezérlőegysége párhuzamosan dolgozza fel a gépi szó összes bitjét. – 2. →Multiprocesszoros számítógép, amely egyidejűleg hajtja végre egy adott program részeit, felhasználva az azok közötti relatív függetlenséget.

paraméter (parameter) – 1. Általános értelmezésben valamely rendszer, objektum, jelenség stb. jellemző mennyisége, amelynek meghatározó szerepe van annak leírásában, jellemzésében. – 2. Programozástechnikai értelmezésben az →eljárások közötti kommunikáció megvalósítására használt eszköz. Lehet formális ~, amely az eljárás algoritmusának leírásában használható, és aktuális ~, amely az illető eljárás aktivizálásának időpontjában biztosítja a végrehajtáshoz szükséges értékeket.

paraméterátadás (parameter transmission) – →szubrutinok, →paraméterezett eljárások, →makroutasítások aktivizálásának fontos tevékenysége: a hívott eljárás →formális paramétereinek meg kell feleltetni a hívás →aktuális paramétereit. A hívó utasítás paramétereinek száma, sorrendje és típusa meg kell, hogy egyezzen a hívott eljárás paramétereinek jellemzőivel. Bármely helyes aritmetikai v. logikai kifejezés csak akkor lehet aktuális paraméter, ha a neki megfelelő formális paraméter skaláris változó. A ~nak három alapvető formája ismeretes: a →cím szerinti hivatkozás, a →név szerinti hivatkozás és az →érték szerinti hivatkozás.

paraméterezett eljárás (parametric procedure) – olyan előre programozott eljárás, amelynél ismert a feldolgozó művelet, de nem ismeretes a végrehajtás konkrét időpontja és a műveletet vezérlő, ill. abban részt vevő operandusok értéke. Ezek konkrét, aktuális értékei csak közvetlenül az eljárás aktivizálása előtt válnak ismertté és helyettesítik a →formális paramétereket. A →programgenerátor kiindulási egységei pl. ~ok.

paraméterlista (parameter list) – magas szintű →programozási nyelvekben használt szintaktikai szerkezet: a hívott eljárás leírásában használt →formális paraméterek, ill. az →eljáráshívásnál megjelenő →aktuális paraméterek rendezett, szövegszerű felsorolása. A formális, ill. aktuális ~knak kompatibiliseknek kell lenniük, ami azt jelenti, hogy az aktuális, ill. formális paraméterek meg kell hogy egyezzenek számban, sorrendben és típusban.

parancsállomány (command file) – felhasználói parancsokat tartalmazó →[adat]állomány, amelyet a →párbeszédes üzemmódban dolgozó →operációs rendszer úgy tud feldolgozni, mintha azokat a felhasználó →terminálon keresztül adta volna meg.

parancsértelmező (command interpreter) – az →operációs rendszerek része, amely a felhasználók utasításait felismeri és az igényelt →erőforrásokat rendelkezésre bocsátja, a megfelelő rendszerprogramok végrehajtását megindítja.

parancsláncolás (command chaining) – →csatornaprogramokban használható módszer, amelynek segítségével lehetővé válik olyan sorozatok (→b/k műveletek) kezdeményezése, mint pl. a többszörös nyomtatás, beolvasás stb. Tkp. a ~ révén képes az →adatcsatorna több utasításból álló csatornaprogramot végrehajtani (szemben az →adatláncolással).

parancsnyelv (command language) – speciális →programozási nyelv a felhasználó és a számítógéprendszer közötti információcsere megvalósítására. Általa érik el a rendszer →erőforrásait, az →operációs rendszer szolgáltatásait, parancsaival állíthatják munkába a →rendszerprogramokat, →segédprogramokat. Jellegzetes ~ a →kötegelt feldolgozásnál használt →munkavezérlő nyelv. Lehetőségeit tekintve elég szegényes nyelv, hiányoznak belőle a magas szintű nyelvekre jellemző →[adat]típusok, az összetett vezérlési szerkezetek stb. Modern változata lehet grafikus is.

párbeszédes fordítás (conversational compiling) – a →forrásnyelvű programok →fordításának →párbeszédes üzemmódban történő végrehajtása. Lényege, hogy a felhasználó az egész fordítás alatt közvetlen kapcsolatban van a számítógéppel, ami lehetővé teszi javítások, módosítások, különböző ellenőrzések azonnali végrehajtását. Elsősorban új programok belövési fázisában előnyös, mivel a →fordítóprogram által jelzett hibák azonnal javíthatók, s a fordítás rögtön újrakezdhető.

párbeszédes terminál (conversational terminal) – →párbeszédes üzemmód megvalósítására kifejlesztett, a központi számítógéppel közvetlenül v. →adatátviteli hálózaton kapcsolódó →terminál. A párbeszéd lebonyolítására billentyűzettel és kijelzővel (képcsővel v. nyomtatóval) van ellátva.

párbeszédes üzemmód (conversational mode) – →időosztásos rendszerekre jellemző üzemmód, amelynek alapvető jellegzetessége a felhasználók és a számítógéprendszer közötti állandó, többszörös üzenetváltás formájában megvalósuló kapcsolat. Miután a kapcsolódó v. kihelyezett állomásoknál (→terminál) ülő felhasználók bebillentyűzték feladatukat v. annak kezdő sorát, a központi számítógép azonnal nyugtázza a vételt, információt közöl arról, hogy mennyi időt kell várni, v. azonnal megkezdi a program végrehajtását, közli az esetleg felfedezett hibákat és az eredményeket. A felhasználóknak lehetőségük van programjaik végrehajtása közben bármikor beavatkozni, menetközi javítások, módosítások eszközölhetők (→lépésenkénti végrehajtás, →ellenőrzött végrehajtás). Legfontosabb újítása a →kötegelt feldolgozással szemben, hogy újra emberközelbe hozta a számítógépet. A mikrogépek jellegzetes üzemmódja.

párhuzamos átvitel (parallel transmission) – →bitpárhuzamos átvitel

paritás (parity) – bináris vektorok attribútuma, amely a vektorban szereplő 1-es bitek számának páros v. páratlan számára utal. Az adatátvitelben használt →hibakorlátozó kódolásban játszik szerepet.

paritásbit (parity bit) – valamely bitvektorhoz rendelt hibajelző bit, amely a bitvektor átvitelének helyességét ellenőrzi. Értékét a bitvektorban helyet foglaló bitpozíciók modulo-2 összegeként határozzák meg, s ez 0 v. 1, attól függően, hogy az összeg páros v. páratlan.

paritás-ellenőrzés (parity check) – a legegyszerűbb →hibakorlátozás, amely csupán páratlan számú hiba jelzésére alkalmas. Általában csak szöveges információk átvitelénél nyújt elegendő védelmet.

paritáshiba (parity error) – adatátvitelben előforduló hiba, ami abban nyilvánul meg, hogy a →paritásbit tartalma nem egyezik meg a hasznos bitvektor →paritásával.

partíció (partition) – 1. Az →operatív memóriának mint →erőforrásnak megosztott használatára vonatkozó kifejezés: az operatív memóriában egyidejűleg jelen lévő programokhoz rendelt memóriaterület elnevezése. Minden program csak a hozzárendelt ~ címhatárain belül férhet a memóriához, s minden rajta kívül eső területet védeni kell tőle. Rendszerint egy-egy hardverregiszter tartalmazza a ~ alsó, ill. felső határát, s minden memóriahozzáférési művelet ezeken a határokon kívül →csapdát eredményez, az illető program végrehajtása felfüggesztődik.

A ~k használata jelenti a →multiprogramozás egyik módszerét. A legegyszerűbb megoldás a rögzített ~k módszere, ahol mindegyik ~ előre meghatározott mérettel és →prioritással rendelkezik. A ~k között van legalább egy →háttérpartíció, ahol →fordítóprogramok, →kapcsolóprogramok, →betöltőprogramok stb., tehát olyan rendszerprogramok dolgoznak, amelyek láncolva, az átfutási időre kevésbé érzékenyen, alacsony prioritással hajtódnak végre. A többi ~ →előtérpartíció típusú, amelyekben a fontosabb, gyors válaszidőt követelő programok v. a rövid futási időt igénylő programok dolgozhatnak.

A ~khoz rendszerint megfelelő típusú →periferikus művek vannak már eleve hozzárendelve, s ez lehetővé teszi, hogy az egyes programokat erőforrásigényük alapján csoportosítsák és a megfelelő ~ba töltsék (→kötegelt feldolgozás).

A →rögzített partíciók módszere elég merev rendszert biztosít. Sokszor dinamikusabb a →változó partíciók módszere, amelynél a ~k számát, nagyságát és prioritási szintjét mindig a pillanatnyi igényeknek megfelelően,

dinamikusan a →felügyelőprogram határozza meg.

2. A személyi számítógépeknél a merevlemez felosztása. Egy partíció az operációs rendszer szempontjából önálló lemezként viselkedik.

particionált [adat]állomány (partitioned file) – olyan →logikai [adat]állomány, amely sorosan szervezett rész[adat]állományokból épül fel, s ezek tartalomjegyzék alapján közvetlenül címezhetők (tehát csak címezhető →adathordozón valósítható meg). Átmenet a →szekvenciális [adat]állomány és a →direkt szervezésű [adat]állomány között, ugyanis a közvetlenül címezhető részállomány csak szekvenciálisan dolgozható fel. A ~ a →könyvtárak tipikus szervezési formája.

particionált adatbázis (partitioned data base) – az →elosztott feldolgozás körülményei között kialakított →adatbázis. Megvalósítása ott lehetséges, ahol a felhasználói igények területileg csoportosíthatók. Elsődleges célja az, hogy a →fizikai adatbázisban tárolt adatokat a felhasználáshoz a lehető legközelebb vigye, megtartva az adatelemek egyszeres tárolását. Előállítása a fizikai adatbázis területi megosztásán alapszik, meghagyva az egységes →logikai adatbázist (→elosztott adatbázis).

particionált alap-adatelérési technika (basic partitioned access method) – →particionált [adat]állományok feldolgozására alkalmazott →alap-adatelérési technika.

particionált memóriakezelés (partitioned memory management) – a többfelhasználós környezet kiszolgálását megvalósító →multiprogramozás egyik szervezési módszere. Az →operatív memória osztott felhasználásán alapszik, két alapvető formája a →rögzített partíciók módszere és a →változó partíciók módszere.

Pascal – az →erősen típusos nyelvek első képviselője, magas szintű algoritmikus →programozási nyelv. 1968-ban N. Wirth elsősorban oktatási célokra tervezte. Közérthetősége, egyszerűsége, ugyanakkor tudományos megalapozottsága és viszonylag könnyű megvalósíthatósága révén igen elterjedt, elsősorban a →mikroprocesszoros rendszerek (→személyi számítógépek) területén. Népszerűségét növelte a Jensen-Wirth szerzőpáros által 1974-ben megjelentetett pontos nyelvleírás. Végleges formáját 1982-ben szabványosította az ISO. Mind az →adatstruktúrák definiálásában, mind a →strukturált programozásban használt vezérlőszerkezetek megvalósításában hatékony, könnyen használható eszközöket nyújt a programozóknak. Magas szinten biztosítja a →rekurzivitást. Moduláris felépítése, blokkstruktúrája szinte kényszeríti a programozót arra, hogy jól strukturált, magas hatásfokú programokat írjon.

A ~ az első olyan nyelv, amely ténylegesen gyakorlatba ültette az →[adat]típus fogalmát, s így kiindulópontja a legmodernebb algoritmikus nyelveknek (→Modula, →CHILL, →Ada).

passzív csillag (passive star) – olyan →csillaghálózat, amelynél a központi csomópont nem végez feldolgozási műveleteket, feladata csupán a külső csomópontok közötti összeköttetés megteremtése.

PC (Personal Computer) – az IBM első termékcsaládja a →személyi számítógépek területén, amely egyhamar a kategória szabványaként is elismerést nyert, számos cég egész terméklistáját a ~vel teljes →kompatibilitást biztosító eszköz tölti ki. Az INTEL →mikroprocesszoraira épül (I8086, I8088, I80186, I80286, I80386), →operációs rendszere az →MS-DOS (újabban a DrDOS és a XENIX).

PC/AT (Personal Computer Advanced Technology) – IBM →személyi számítógép, az INTEL 80286 →mikroprocesszorra alapozva. →Operációs rendszere az →MS-DOS. A →PC/XT továbbfejlesztett változata.

PC-DOS – az →MS-DOS →operációs rendszerrel azonos rendszer az IBM →személyi számítógépein.

PC/XT (Personal Computer eXTended) – IBM →személyi számítógép, amely eredeti kiépítésben tartalmaz egy →merevlemezes egységet. Az INTEL 8086 →mikroprocesszorára épül, →operációs rendszere az →MS-DOS.

periféria (peripheral equipment) – valamely számítógéprendszernek a →központi vezérlőegységtől és az →operatív memóriától különböző alegységeit összefoglaló gyűjtőfogalom. Ide tartoznak az összes →háttértárak, →bemeneti egységek, →kimeneti egységek

és általában mindazok a berendezések, amelyek a számítógéprendszer és környezete között kapcsolatot teremtenek.

periferikus egység (peripheral unit) – →periferikus mű

periferikus mű (peripheral device) – a számítógéppel összekapcsolt, vele együttműködő berendezés, amelynek működését a →központi vezérlőegység vezérli ugyan, de rendszerint nem közvetlenül, hanem külön saját vezérlőegységen keresztül (→adatcsatorna). Ilyenformán a ~ működése és a központi vezérlőegység működése között párhuzamosság biztosítható, amely a két működési sebesség közötti lényeges különbséget hivatott kiegyenlíteni.

permanens memória (permanent/fixed memory) – olyan →memória, amelynek rögzített tartalma nem írható felül a szokásos beírási műveletekkel, tehát tartalmát csak kiolvasni lehet (így az adatok beírására szolgáló elektronikára nincs is szükség). Tartalmát az előállító v. a felhasználó rögzítheti speciális berendezés segítségével. Ez utóbbi esetben →programozható permanens memóriáról beszélnek (szemben a huzalozott memóriával, amelynek tartalma egyáltalán nem változtatható). Legmodernebb kivitelezése az →újraprogramozható permanens memória, amelynek tartalmát a felhasználó (→off-line üzemben) törölheti és újra rögzítheti. Elsősorban a →mikroprogramozás céljára kifejlesztett memóriatípus, legfontosabb jellemzője az igen rövid elérési idő, az alacsony bitköltség, valamint az eredeti tartalom biztonsága.

piktogram (pictogram) – →ikon

PILOT (Programmed Inquiry, Learning Or Teaching) – a →számítógéppel segített oktatás céljaira kifejlesztett speciális →programozási nyelv.

PL/1 (Programming Language One) – igen elterjedt magas szintű →programozási nyelv. Megalkotói igyekeztek egyetlen nyelvben összpontosítani mindazokat az előnyöket, amelyeket a →COBOL, ill. a →FORTRAN és az →ALGOL nyelvek biztosítottak. A COBOL-ból a nagy adathalmazok kezelésére kifejlesztett hatékony eszközöket, a FORTRAN-ból a matematikai formulák, algoritmusok kódolására alkalmas eszközöket, míg az ALGOL-ból a blokkszerkezetet tartották szem előtt. Így lett a ~ általános célú, eljárásra orientált, blokkszerkezetű, magas szintű programozási nyelv, amit a lehető legszélesebb problémakörökben lehet alkalmazni. Moduláris felépítése lehetővé teszi, hogy az egyes felhasználók csak a tényleges igényeiket megvalósító résznyelvet használják. A moduláris felépítés, valamint a nyelv alapkoncepciója lehetőséget ad a teljes nyelvből önállóan zárt részhalmazok, különböző nyelvi szintek létrehozására. Bár magas fokú gépfüggetlenséget biztosít, megfelelő eszközökkel rendelkezik az →operációs rendszerrel történő hatékony együttműködésre. Lehetőséget ad a felhasználóknak a →csapdák, →megszakítások kezelésére, a →b/k műveletek közben előfordulható hibák felismerésére és feldolgozására stb.

A ~ alapvető erőssége a blokkszerkezet: minden program egy v. több blokkból áll, amelyek között mindig van egy főeljárás. Ugyancsak a blokkszerkezet teszi lehetővé a →dinamikus memóriakiosztást, a →dinamikus változók használatát, vagyis azt, hogy a változók számára történő memória-hozzárendelést a felhasználó felügyelete alá helyezze.

PL/M (Programming Language for Microprocessors) – a →PL/1-hez hasonló, blokkszerkezetű, magas szintű →programozási nyelv, amelyet az INTEL cég fejlesztett ki 8 bites →mikroprocesszoraira, majd adaptált a 16 bites mikroprocesszorokra is.

polinom-ellenőrzés (polynomial check) – →adatátvitel helyességének ellenőrzésére használt módszer, amikor az átvitelre →ciklikus kódot alkalmaznak. Lényege, hogy a vételezett kóddal (polinomként értelmezve) és az ismert generáló polinommal meghatározott műveleteket végeznek, az eredményt pedig összehasonlítják az ugyancsak vételezett eredménnyel (→blokkellenőrző karakter). Eltérésnél az információ újraközvetítését kérik.

ponttól pontig összeköttetés (point-to-point connection) – →adatátviteli hálózatokban két →csomópontot összekötő →kommunikációs csatorna.

postfix forma (postfix form) – aritmetikai kifejezések felírásának olyan formája, amelyben a műveleti jelet az operandusok után helyezik. Fordított →lengyel forma néven is is-

meretes, ugyanis J. Lukasiewicz lengyel matematikus használta először. Alapvető jelentősége az, hogy elhagyható a zárójelek használata, a műveletek elvégzési sorrendje mégis meghatározott. Pl. az (a + b) * (c – d) aritmetikai kifejezés ~ja: ab + cd – *.

„post-mortem" program (post-mortem program) – →operációs rendszerhez tartozó program, amelynek az a feladata, hogy →szoftver v. →hardver típusú →végzetes hiba után a munkaregiszterek és az →operatív memória tartalmát egy előre meghatározott →háttértárba kimentse. A felhasználó által esetleg igényelt →memóriakiíratás (post-mortem dump) ebből a háttértárból lehetséges. Fontos eszköz a programok belövésének és tesztelésének időszakában.

precedenciaszabály (precedence rule) – aritmetikai műveletek végrehajtási sorrendjét meghatározó egyezmény. Eszerint elsődleges a hatványozás, másodlagos a szorzás és az osztás, harmadlagos az összeadás és a kivonás. Ha azonos precedenciájú műveleti jel van az operandus bal és jobb oldalán, akkor az operandus a tőle balra lévő műveleti jelhez fog tartozni (→„balról jobbra" szabály). Számítástechnikai kiterjesztése a logikai és relációs műveletek sorrendjét is szabályozza: elsődlegesek a relációs kifejezések, másodlagos a →negáció, harmadlagos a →logikai szorzás, negyedleges pedig a →logikai összeadás. Megfelelő zárójelezéssel természetesen befolyásolni lehet a műveleteknek ezt az előírt sorrendjét.

Ha a logikai kifejezések magukban foglalnak aritmetikai, relációs és logikai kifejezéseket is, akkor tekintet nélkül az egyes operandusok kifejezésbeli helyzetére, a végrehajtás sorrendje a következő: aritmetikai műveletek kiértékelése, relációs kifejezések kiértékelése, negáció, logikai szorzás, logikai összeadás.

predikátum (predicate) – a →matematikai logika alapvető fogalma. Olyan egy- v. többváltozós kijelentő mondat, amely – a változók behelyettesítése után – érvényes →ítéletet jelent. Pl. x>3 egyváltozós ~, (x>3) ÉS (y<x) kétváltozós ~.

Szorosan kapcsolódó fogalom a kvantor. Ha x adott halmaz tetszőleges eleme és $\pi(x)$ tetszőleges ~, akkor:

1) $\exists x \cdot \pi$ – vagyis létezik legalább egy olyan x, hogy π igaz legyen; ez az ún. egzisztenciális kvantor;

2) $\forall x \cdot \pi$ – vagyis π igaz, bármilyen behelyettesítést használna is x számára; ez az ún. univerzális kvantor.

prefix forma (prefix form) – aritmetikai kifejezések felírásának olyan formája, amelyben a műveleti jelet az operandusok elé helyezik. Akárcsak a →postfix forma, →lengyel forma néven is ismeretes, ugyanis J. Lukasiewicz lengyel matematikus használta először. Alapvető jelentősége, hogy elhagyható a zárójelek használata, a műveletek elvégzésének sorrendje így is meghatározott. Pl. az (a + b) * (c – d) aritmetikai kifejezés ~ja: * – ab – cd.

primitív [adat]típus (primitive [data] type) – a →kiterjeszthető nyelvekkel kapcsolatos fogalom. Olyan elemi →[adat]típus, amely jelen van az illető nyelv kiindulási magjában, s további [adat]típusok megvalósítására (definiálására) használható.

prioritás (priority) – →megszakítási rendszereken alapuló számítógéprendszereknél a →munka v. az ahhoz tartozó →feladatok (taszkok) fontosságának kifejezésére használt jellemző, amelyet a felhasználó v. a számítógéprendszer üzemeltetője határoz meg, s amelynek alapján a →multiprogramozás megvalósítható. Két, azonos időben érkező →megszakítás fellépésénél a nagyobb ~sal rendelkező részesül előnyben. Hasonlóképpen, ha több program van egyszerre abban az állapotban, hogy megkaphatná a vezérlést, a nagyobb ~sal rendelkező program fog először hozzájutni a →központi vezérlőegységhez.

Ha egy rendszerben →valós idejű programok is üzemelnek, ezeknek mindig a legnagyobb szintű ~sal kell rendelkezniük.

prioritáskiosztás (priority allocation) – →multiprogramozásos rendszerekben használt stratégia, amelynek lényege az, hogy az egy időben végrehajtás alatt lévő programokhoz bizonyos elsőbbségi tulajdonságot, →prioritást rendelnek annak függvényében, hogy az illető feladat mennyire idő- v. →erőforrásigényes, ill. mennyire fontos az egész rendszer megfelelő működése szempontjából. Az adott prioritási szint egyben azt is meghatározza, hogy az illető program végre-

hajtását milyen jellegű külső esemény függesztheti fel, ill. a felfüggesztett állapotból milyen körülmények között kerülhet újra aktív állapotba.

Statikus ~nál a rendszerben egy időben jelen lévő programoknak már eleve meghatározott prioritási szintje van, s ez nem változtatható meg, csak →gépkezelői beavatkozás útján. Dinamikus ~nál a →felügyelőprogram időről időre megváltoztathatja a programok prioritási szintjét, azok pillanatnyi erőforrás-igénye, ill. a rendelkezésre álló erőforrások mennyisége és minősége alapján.

privát [adat]típus (private [data] type) – az →absztrakt [adat]típust, ill. az →információrejtést megvalósító programozási kategória az →Ada nyelvben. Olyan típusdefiníció, amelyet →csomagba rejtve bocsátanak a program többi részének rendelkezésére. Az értékkészlet ábrázolására, ill. az elvégezhető műveletekre vonatkozó implementálási részletek rejtve maradnak, s így azoktól független programokat lehet írni. Az ilyen programok változatlanok maradhatnak akkor is, ha a későbbiekben a ~ konkrét megvalósítási részleteit megváltoztatják.

privilegizált állapot (privileged state) – a →központi vezérlőegység olyan állapota, amikor a teljes →utasításkészlet (beleértve a →privilegizált utasításokat is) elérhető, felhasználható (nevezik még felügyelő állapotnak is).

privilegizált utasítás (privileged instruction) – valamely számítógép →utasításkészletéhez tartozó olyan utasítás, amely kizárólag egyes →rendszerprogramok számára van fenntartva. Ilyen típusú utasítások a memóriavédelemmel és a memóriakiosztással, a →megszakítások feldolgozásával kapcsolatos utasítások (pl. a →programállapotszó megmentésére), az →állapotregiszterekkel dolgozó, a →periferikus műveket vezérlő utasítások (→csatornaprogramok) stb. A felhasználói programokban megjelenő ~ kódját az →operációs rendszer ismeretlen utasításkódként kezeli, s →csapdát jelent.

problémaorientált nyelv (problem-oriented language) – magas szintű →programozási nyelv, amelyet meghatározott típusú feladatok programozására fejlesztettek ki. Pl. tudományos-műszaki feldolgozásokra (→ALGOL, →BASIC, →FORTRAN, →PL/1 stb.), gazdasági-kereskedelmi adatfeldolgozásra (→COBOL, →PL/1, RPG stb.), listafeldolgozásra (→LISP, MLISP, QLISP stb.), szimbolikus-logikai feldolgozásokra (→Prolog, QLISP stb.), rendszerprogramozásra (→B, →BCPL, →C, →CDL, XPL), szimulációra (→SIMULA, →SIMSCRIPT stb.), hardvertervezésre (→CHDL) stb.

procedurális nyelv (procedural language) – →eljárásra orientált nyelv

processzor (processor) – rendszerint elektronikus berendezés, amelynek alapfeladata valamely rendszer működéséhez szükséges vezérlőjelek előállítása, az egész rendszer vezérlése. Címképzéssel utasításokat képes kiolvasni a →memóriából, értelmezi és végrehajtja azokat. Ilyen alapon ~ a →digitális számítógép →központi vezérlőegysége, a →b/k műveleteket vezérlő →adatcsatorna, a →mikroprocesszor stb. Tágabb értelmezésben ~ a programozási nyelveket értelmező →nyelvi processzor is.

program (program) – 1. Statikus értelmezésben valamely →algoritmusnak, →eljárásnak →gépi utasításokkal v. →programozási nyelvek utasításaival történő ábrázolása, leírása, kódolása. – 2. Dinamikus értelmezésben utasítások időben lefolyó, szekvenciális végrehajtása, aminek eredménye – adott bemenő adatok alapján – a kívánt kimenő adatok előállítása.

programállapotszó (program status word) – →multiprogramozásos rendszerekben az →operációs rendszer által használt információvektor, amely az adott program végrehajtására v. felfüggesztés utáni újraindítására vonatkozó információkat tartalmazza. Az adatok, amelyeket rendszerint egybegyűjt, a következők: a következő végrehajtandó utasítás címe, a programhoz rendelt →prioritás, a tárhozzáférési kulcs, a →csapdákhoz rendelt →maszkok állapota, a programhoz rendelt →partíció száma stb. A különböző partíciókban végrehajtás alatt lévő mindegyik program saját ~val rendelkezik, amelyet az operációs rendszer utal ki, s ugyancsak ő módosíthat. Valahányszor a program végrehajtása időlegesen felfüggesztődik, az operációs rendszer a megfelelő ~ban helyezi el mindazokat az információkat, amelyekre a program újraindításához szükség van.

programáthelyezés (program relocation) – →áthelyezés

programblokk (program block) – bizonyos magas szintű →programozási nyelveken (pl. →ALGOL, →PL/1, →Pascal stb.) kódolt programok olyan alegysége, amely a benne definiált →azonosítók hatáskörét kijelöli. A ~ban lokálisnak deklarált változók azonosítói csak az adott ~ határain belül értelmezettek. Hierarchikusan egymásba skatulyázott ~oknál a külső ~ azonosítói globális jellegűek a belső blokkok számára, vagyis a deklarációt tartalmazó blokkon kívül is értelmezettek. A definíciókat rekurzív módon kell értelmezni, mert a lokális változókhoz csak az illető ~ végrehajtásának idejére rendelnek tényleges memóriaterületet.

programcsomag (program package) – általános jellegű, legtöbbször paraméterezhető programok gyűjteménye, amely adott problémakör – bizonyos szempontok szerint egységbe foglalt – feladatainak megoldására alkalmas. Esetleg forrásnyelvű, de legtöbbször végrehajtható formában áll a felhasználók rendelkezésére valamilyen mágneses →adathordozón.

Abszolút formájú a ~, ha különböző felhasználók minden változtatás nélkül használhatják ugyanannak a konkrét feladatnak az elvégzésére. Legtöbbször mégis paraméterezett formájú a ~, amelynél a különböző felhasználók saját szükségleteiknek megfelelően paramétereken keresztül válogathatják ki a bemenő, ill. kimenő adatok formáját v. a ~ által rendelkezésre bocsátott funkciók közül a speciális körülményeknek megfelelőt használhatják.

programegység (program unit) – →forrásnyelvű v. →tárgynyelvű utasítások sorozata, amely adott feladat megoldása szempontjából egy részfeladatot megoldó logikai egységet alkot. Nem tévesztendő össze a →modul fogalmával, amely az adott program – esetleg ~ – szerkezeti szempontjából alkot önálló egységet. Más megközelítésben valamely programtermék funkcionálisan v. logikai szempontból független része, s mint ilyen nem tévesztendő össze a →fordítási egységgel, ugyanis két v. több ~ is képezhet egyetlen fordítási egységet.

programérvényesítés (program validation) – valamely programtermék logikai helyességének és a specifikációkban foglalt előírásoknak a bizonyítása valódi környezetben, előre meghatározott tesztprogramok segítségével.

programfuttatás (program run/execution) – valamely →program utasításai által előírt műveletek végrehajtása konkrét számítógép →erőforrásainak felhasználásával. Tkp. az adott program egyszeri teljes végrehajtását jelenti.

programgenerátor (program generator) – speciális megvalósítású és célú →segédprogram. Feladata az, hogy előre elkészített →paraméterezett eljárásokból a felhasználó konkrét feladatát megoldó programterméket állítson elő. Jellegzetes képviselője a →rendezőprogramot előállító ~, és hasonló logikájú programtermékkel szokás elvégezni a →rendszergenerálást is.

programhitelesítés (program certification) – valamely programtermék logikai helyességének és a specifikációkban foglalt előírásoknak a jóváhagyása a hosszabb idő alatt összegyűlt pozitív futtatási tapasztalatok alapján.

programigazolás (program verification) – valamely programtermék logikai helyességének és a specifikációkban foglalt előírásoknak a bizonyítása megfelelően kiválasztott tesztkörnyezetben.

programkönyvtár (program library) – →könyvtárban összegyűjtött, már kipróbált programok rendszere, amelyeket későbbi felhasználás céljából valamilyen mágneses →adathordozón tároltak. A tárolási forma szerint a ~ lehet →forrásnyelvű, →tárgynyelvű v. közvetlenül végrehajtható. A ~ban tárolt programok (programegységek, programmodulok stb.) bármikor beépíthetők újabb programtermékekbe. A ~ak előállítására, kezelésére, módosítására, esetleg törlésére általában speciális rendszerprogram, a →könyvtárkezelő program áll a felhasználók rendelkezésére.

programmegszakítás (program interrupt) – az éppen aktív állapotban lévő program végrehajtásának ideiglenes felfüggesztése (esetleg végleges megállítása) valamilyen külső esemény hatására v. valamilyen →végzetes hiba v. →csapda következtében. A →megszakítást az →operációs rendszer megfelelő szubrutinja dolgozza fel oly mó-

don, hogy a megszakítási jel hatására a futó program végrehajtása felfüggesztődik, és a vezérlés automatikusan a megszakítás okát felderítő és annak megfelelően intézkedő szubrutinnak adódik át.

~kor speciális memóriarekeszek tárolják a felfüggesztett program állapotát (→programállapotszó) az esetleges későbbi folytatás céljából. A ~t okozó jel jöhet egy-egy →periferikus műtől (jelezve pl. az adatcsere befejezését), a gépkezelőtől v. nagyobb →prioritással rendelkező programtól, de belső végzetes hiba (ismeretlen utasításkód, hibás cím, védett memóriához való hozzáférési próbálkozás stb.), esetleg órajel is okozhat ~t.

programoptimalizálás (program optimization) – számítástechnikai értelmezésben a programtermékek olyan tervezése, ill. változtatása, hogy azok megközelítsék a maximális hatékonyságot. Ez rendszerint olyan egészséges kompromisszum megvalósítását jelenti, amely a minimális →operatív memória, ill. →adathordozó felhasználása mellett a leggyorsabb feldolgozási sebességet biztosítja. Különleges jelentősége van a →rendszerprogramok megvalósításában, tekintettel arra, hogy ezek színvonala erősen befolyásolja az egész számítógéprendszer hatékonyságát.

programozás (programming) – szűkebb értelemben valamely feladatot megoldó eljárás műveleteinek →gépi utasítások v. →programozási nyelvek utasításai segítségével történő leírása adott számítógépen való végrehajtás céljából. Ez a meghatározás tkp. csak a kódolásra vonatkozik, a ~ ennél sokkal komplexebb tevékenység. Tágabb értelmezésben az az összetett munkafolyamat, amelynek célja és eredménye programtermékek előállítása, s amely több fontos tevékenységet tartalmaz: a megoldandó feladat definiálása, a megfelelő eljárás (algoritmus) kiválasztása v. megtervezése, az eljárás logikai szerkezetének megtervezése, a szükséges →hardver- és →szoftverkörnyezet megállapítása, az eljárás szerkezeti lebontása, a megfelelő programozási nyelv meghatározása, az eljárás kódolása, a programegységek (ill. a program) számítógépre vitele és belövése, a programtermék érvényesítése, dokumentációjának elkészítése, üzembe helyezése és karbantartása.

A ~ rendszerezésére, szabványosítására többfajta módszertant javasoltak, gyakorlati értéke közülük a →moduláris programozásnak és a →strukturált programozásnak van.

programozási hiba (programming error) – valamely eljárás v. algoritmus számítógépre vitele során elkövethető hiba. Lehet →szintaktikai hiba, →szemantikai hiba (ezek az illető programozási nyelv nem kielégítő ismeretéből fakadnak), →módszerhiba (okozója a valamilyen szempontból nem megfelelő algoritmus) és →logikai hiba.

A ~k külön osztályát alkotják azok, amelyek a kódolási tevékenység felületességéből fakadnak. Ilyenek: 1) az aritmetikai műveletekkel kapcsolatos hibák (nullával való osztás, →túlcsordulás, →alulcsordulás, nem megfelelő →[adat]típus használata, hibás zárójelezés, szükséges zárójelek elhagyása, kezdőérték nélküli változók használata stb.); 2) a →jelsorozatokra vonatkozó hibák (a szükségesnél rövidebb v. hosszabb láncok definiálása, a sorozat határait meghaladó hivatkozások stb.); 3) a →tömbökre vonatkozó hibák (nem megfelelő méretű tömbök deklarálása, indexhatárok felcserélése, tömbhatárok túllépése stb.); 4) a ciklusszervezésre vonatkozó hibák (a ciklusváltozó kezdeti és végértékének hibás beállítása, a lépésköz helytelen megválasztása, a ciklus kilépési feltételeinek hibás meghatározása stb.); 5) az adatcserével kapcsolatos hibák (a szükségesnél kevesebb v. több adat beolvasása, hibás formátumok, sablonok használata, a →blokkolási tényező hibás megválasztása és használata, a →pufferek hibás kezelése stb.); 6) a →szubrutinokra, a →függvényekre vonatkozó hibák (hibás paraméterlista használata, →paraméterátadás hibás kezelése stb.).

programozási környezet (programming environment) – technológiai programgyűjtemény, amelynek központjában rendszerint meghatározott →programozási nyelv áll. A programgyűjteményben szereplő összes program az illető programozási nyelven történő programfejlesztést segíti elő. A minimális ~ rendszerint a következő típusú programokat foglalja magában: az illető programozási nyelv →fordítóprogramját (esetleg →értelmezőprogramját), a →kapcsolóprogramot, a →betöltőprogramot, a →forrásprogramok ill.

→tárgyprogramok kezelését ellátó →könyvtárkezelő programot, a forrásnyelvű programok előállítását támogató →szövegszerkesztő programot stb.

A ~ fogalmát tkp. az →Ada programozási nyelv előállítói vezették be, bár valami hasonlót jelentett az →UNIX →operációs rendszer is, amely a →C programozási nyelven alapszik. Az APSE – vagyis az Ada Programming Support Environment – lényege az, hogy maga az Ada nyelv és a különböző programfejlesztő eszközök egységes egészt alkotnak. Igen látványos és hatékony ~eket a Borland cég fejlesztett ki TURBO családnévvel (a →BASIC, a →Pascal, a →C, ill. a →Prolog nyelvek részére).

programozási nyelv (programming language) – bármilyen mesterséges nyelv, amely megfelelő eszközökkel rendelkezik adatstruktúrák és az azokat kezelő, átalakító eljárások egyértelmű leírására. A gyakorlati meghatározáshoz hozzátartozik még egy fontos követelmény: létezzen egy olyan eszköz, amely az illető ~ struktúráit felismeri és a számítógépen feldolgozhatóvá teszi (ez az eszköz végeredményben az illető ~ →fordítóprogramja v. →értelmezőprogramja). A ~nek jól kidolgozott elmélete van (→formális nyelvek).

A ~ definiálásához általában a →szintaxis és a →szemantika leírása szükséges, gyakorlati felhasználásához viszont egy-egy konkrét számítógéprendszeren megvalósított implementáció speciális felhasználói dokumentációja segít.

A ~ek több szempontból osztályozhatók. Konkrét számítógéptől való függőségük szempontjából lehetnek →gépi nyelvek, →asszembler nyelvek v. →magas szintű programozási nyelvek. Felhasználási területük szerint lehetnek →eljárásra orientált nyelvek és →parancsnyelvek, →szimulációs nyelvek, →szimbólumfeldolgozó nyelvek stb.

programozási segédlet (programming aids) – programtermékek tervezését, előállítását, belövését, tesztelését, karbantartását stb. elősegítő segédeszközök összefoglaló elnevezése. A legfontosabb ilyen rendeltetésű eszközök, amelyek rendszerint rendelkezésre állnak egy-egy számítógéprendszer felhasználói számára, a →magas szintű v. →asszembler nyelvek és azok →fordítóprogramjai (→értelmezőprogramjai), a →rendszerprogramok (→kapcsolóprogramok, →betöltőprogramok), a →kezelőprogramok (→könyvtárkezelő programok, →rendezőprogramok, →szövegszerkesztők stb.), a →tesztgenerátorok stb. Modern, integrált változata a →programozási környezet.

programozási technológia (software engineering) – a számítástechnika egyik fontos kutatási területe, amely a programtermékek nagyüzemi, ipari jellegű előállításával foglalkozik. Elsőrendű célja olyan eszközök és módszerek tanulmányozása és előállítása, amelyek a gazdasági hatékonyság keretein belül a specifikációkat legjobban megvalósító, gyakorlatilag hibamentes programtermékek előállítására alkalmasak. Legfontosabb gyakorlati eredményei a →moduláris programozás és a →strukturált programozás.

programozható automata (programmable automaton) – olyan →automata, amelynek bemenő információi két csoportba oszthatók: a tulajdonképpeni feldolgozandó adatok és a feldolgozási algoritmust megvalósító program. A ~ állapotait a bemenő adatok, a pillanatnyi állapotok és a programok egyaránt befolyásolják.

programozható megszakításvezérlő (programmable interrupt controller) – →személyi számítógépek →alaplapján található, irányítási feladatokat ellátó →integrált áramkör. A ~ figyeli a →megszakításokat, fogadja a megszakítási jeleket, →prioritásuk alapján meghatározza fontosságukat, majd ennek alapján továbbítja őket a →központi vezérlőegységnek.

programozható periférialillesztő (programmable peripheral interface) – →személyi számítógépek →alaplapján található, irányítási feladatokat ellátó →integrált áramkör. A ~ kapcsolja a →sínrendszerhez a számítógéphez tartozó →periferikus egységeket.

programozható permanens memória (programmable read-only memory) – olyan →permanens memória, amelynek tartalmát a felhasználó rögzíti beépítés előtt, de a rögzített tartalom csak kiolvasásra alkalmas.

programozható terminál (programmable terminal) – saját vezérlőegységgel és memóriával rendelkező →terminál, amelyet a felhasználó igényeinek megfelelően progra-

mozhat. Ide tartoznak az →intelligens terminálok egyes típusai, valamint a →kötegelt feldolgozású terminálok. Saját memóriája révén a ~ egyidejűleg végezhet feldolgozást és adatátvitelt a központi számítógéppel.

programozó (programmer) – 1. Programtermékek előállításán dolgozó szakember, aki a számítógép által elvégzendő műveleteket valamilyen →programozási nyelven megfogalmazza, a programok tesztelését, esetleg karbantartását elvégzi. – 2. Speciális berendezés, amellyel →permanens memóriák, ill. →programozható permanens memóriák tartalmát előállítják.

programozott erőforrás (programmed resource) – valamely →operációs rendszer által nyújtott, a →hardver erőforrásokat kiegészítő szolgáltatás. Ezek a szolgáltatások nagyrészt elektronikus úton is megvalósíthatók, de sokszor gazdaságosabb őket programozottan biztosítani. A ~ok igénybevételének vezérlése az operációs rendszer egyik legfontosabb feladata. Jellegzetes ~ok pl. a →perifériát vezérlő utasítássorozatok (→csatornaprogramok), a programozott úton megvalósított →lebegőpontos műveletek, a →logikai [adat]állományokat kezelő szubrutinok s tágabb értelmezésben a →felügyelőprogram szolgáltatásai.

programozott oktatás (programmed instruction) – korszerű oktatási forma. Lényege az, hogy az elsajátítandó tananyagot megfelelő szempontok szerint elemi részekre osztják, és a tanulókat kérdések és az azokra adott válaszok kiértékelése alapján vezetik rá az ismeretanyagra. A ~ igen jól automatizálható speciális oktatógépek, ill. számítógépek segítségével.

programspecifikáció (program specification) – programtermékek előállításának egyik fontos fázisa, ill. annak terméke. Lényegében mindazt pontosan és kimerítően meg kell határoznia, amit a leendő programnak el kell majd végeznie, figyelmen kívül hagyva annak konkrét megvalósítási módját.

programszakasz (program section) – →szakasz

programszegmens (program segment) – →átfedési szerkezetek funkcionális és szerkezeti egysége (→programszegmentálás). Olyan →szegmens, amely utasításokat és adatokat v. csak utasításokat tartalmaz (szemben az →adatszegmenssel, amely viszont csak adatokat tartalmaz). Az →operatív memóriába való →betöltését rendszerint az →operációs rendszer végzi el automatikusan, az átfedési szerkezet alapján, →asszembler nyelven viszont a felhasználó maga is dönthet betöltéséről.

programszegmentálás (program segmentation) – az →operatív memóriával való gazdálkodás céljából a programnak olyan önálló egységekre történő felbontása, amelyek egyenként tölthetők be a külső memóriából. Egy programban általában több tevékenységet kell meghatározott sorrendben elvégezni, s az azokat megvalósító programrészeket célszerű csak a tényleges felhasználás időpontjában az operatív memóriába betölteni. Ez a módszer jelentős tármegtakarítást eredményezhet, s ugyanakkor elengedhetetlen feltétele olyan hosszúságú programok végrehajtásának, amelyek egyszerre nem lennének betölthetők.

A ~nak két alapvető gyakorlati megvalósítása ismeretes. Klasszikus formájában a programrészek (szegmensek) az →átfedési szerkezet alapján kerülnek az operatív memóriába (a gyökérszegmens a végrehajtás egész ideje alatt a memóriában marad). Ennek a szerkezetnek a meghatározása a felhasználó, tényleges megszerkesztése pedig a →kapcsolóprogram feladata. A ~ modernebb változata a →lapkezelés: itt, a felhasználó beavatkozása nélkül, a programot a speciális kapcsolóprogram osztja szegmensekre és →lapokra, s a lapok csak akkor kerülnek az operatív memóriába, ha közvetlenül hivatkoznak rájuk (→címtartományukra).

programszerkesztő (program editor) – →párbeszédes üzemmódban dolgozó →operációs rendszerekben rendszerint megtalálható segédprogram, amely a →forrásnyelvű programok szövegszerű előkészítését segíti elő. Szövegösszefüggések v. programsorok számának segítségével a szerkesztendő program bármelyik sorát lokalizálhatja, s ott beszúrásokat, javításokat, törléseket végezhet el. Ezeket a műveleteket közvetlenül a forrásnyelvű →könyvtárakban lehet elvégezni, ahonnan a →fordítóprogramok kiolvashatják és lefordíthatják a programokat.

programtesztelés (program testing) – 1. Általános értelmezésben az a tevékenység, amelynek során a programterméket rendszeres próbáknak vetik alá, hogy megvizsgálják, mennyire felel meg bizonyos feladat számítógépes megoldásában. Más szóval: ellenőrzik, vajon a programtermék elvégzi-e mindazokat a tevékenységeket, amelyeket a programtermék specifikációjában előírtak, és ha igen, mennyire helyesen. – 2. Szűkebb értelmezésben az a tevékenység, amelynek célja a programban esetleg előforduló hibák felderítése, lokalizálása és kijavítása. Ezt a szűkebb körű tevékenységet a magyar szakirodalomban belövésnek is nevezik.

programváltozat (program version) – a még fejlesztés alatt álló v. véglegesnek tekintett programtermék olyan szintű megvalósítása, amely úgy áll a felhasználók rendelkezésére, hogy esetleg még tartalmaz bizonyos megkötéseket, de már mentes az előző állapotban észlelt hibáktól v. megszorításoktól. Gyakorlatilag a programtermékek több ~on keresztül nyerik el végső formájukat, egyre jobban megközelítve a specifikációkban meghatározott előírásokat.

programvezérlésű gép (program-controlled machine) – olyan berendezés, amelynek működése – jól meghatározott célok érdekében – utasítások segítségével (vagyis programmal) vezérelhető. A vezérlőprogramot általában valamilyen bemeneti adathordozón (→lyukszalagon, →lyukkártyán stb.) tárolják, de a vezérlőjeleket sokszor szolgáltathatja a számítógép →operatív memóriájában található program végrehajtása is.

Prolog – szimbolikus feldolgozásra kifejlesztett magas szintű →programozási nyelv. A hetvenes évek elején a Horn-formulák alapján dolgozta ki R. Kowalski és P. Hayes a →logikai programozás elméletét, és ennek gyakorlati megvalósításaként készítette el A. Colmerauer 1973-ban az első ~ implementációt. Előnyösen egyesíti magában a matematikai logika és az algoritmikus gondolkodásmód elemeit.

Alapvető jellegzetessége, amely a többi magas szintű programozási nyelv fölé helyezi, hogy →szemantikája pontos, matematikailag megalapozott. Mivel logikára épülő programozási nyelv, igen alkalmas →szakértőrendszerek megvalósítására, →adatbázisok lekérdezési nyelvének integrálására, természetes nyelvek feldolgozására és magasabb szintű következtetési funkciók megvalósítására.

Világszerte jelentősen előmozdította a ~ iránti érdeklődést az a körülmény, hogy az 1981 végén bejelentett japán ötödik generációs számítógéptervezet bázisnyelvként választotta.

protokoll (protocol) – →adatátviteli hálózatokban a kommunikáció kezdeményezéséhez és fenntartásához szükséges, szigorúan kötött szabályok rendszere. Adott hálózatban több szinten is létezhet: összeköttetés-összeköttetés, adatvégállomás-adatvégállomás, ill. előfizető-kapcsoló szinten (→hétréteges referenciamodell). A ~ok határozzák meg az →üzenetek, ill. →adatcsomagok felépítését, ill. lebontását, az adatátviteli hibák javítását, az adatállomások lekérdezésének megszervezését stb.

pszeudokód (pseudocode) – eljárások leírására alkalmas olyan utasítások sorozata, amelyek nem egy konkrét számítógép v. programozási nyelv utasításai. Rendszerint a programozási struktúráknak valamilyen természetes nyelv →szintaxisával történő leírása (pl. „ha...akkor...ellenben" v. „ameddig ... hajtsd végre..." stb.). Elsősorban a programtermékek tervezési és dokumentálási fázisában van jelentősége, ugyanis lehetővé teszi a logikai struktúra pontos, könnyen érthető leírását, valamely programozási nyelven történő kódolását, valamint a logikai terv és a program közötti egyeztetés közérthetőségét.

pszeudonyelv (pseudolanguage) – programfejlesztésben használható olyan jelölésrendszer, amelyet a →strukturált programozás magasabb szintjein lehet használni. Rendszerint egy →absztrakt számítógép nyelvének tekinthető.

pszeudoutasítás (pseudoinstruction) – →asszembler nyelvekre jellemző olyan utasítástípus, amelynek segítségével a →fordítási műveletre vonatkozó információkat leírják. Alapvető tulajdonsága, hogy a fordítás eredményeként előállított →tárgyprogramba közvetlenül nem épül be (nem felel meg neki semmilyen →gépi kód), csupán a hatása jelentkezik. Általában →szakaszok elhatárolására, →bázisregiszterek kijelölésére, szimbólumok definiálására, memóriarekeszek kiuta-

lására, konstansok generálására stb. használható.

pszeudovéletlen számok (pseudorandom numbers) – az $x^1, x^2, \ldots, x^n$ véletlen számoknak olyan sorozata, amely az $x^n = f(x^1, x^2, \ldots, x^n)$ rekurzív összefüggés segítségével állítja elő az egyenletes eloszlású véletlen számokat. Rendszerint statisztikai vizsgálatokkal állapítják meg, hogy a rekurzív összefüggés mennyire felel meg a véletlen számokkal szemben támasztott követelményeknek.

puffer (buffer) – az →operatív memória meghatározott része v. valamilyen kis kapacitású speciális memória, amelyen keresztül megvalósul az operatív memória és valamely →periferikus mű közötti adatcsere. Tekintettel arra, hogy a periferikus műveket rendszerint saját speciális →adatcsatorna vezérli, az átvitel idejére az adatcserét kezdeményező program végrehajtása felfüggesztődik, így a →központi vezérlőegység kiutalható más programoknak, miközben az adatcsere a periferikus mű és a ~ között folytatható. Több ~ segítségével a programfelfüggesztés is kiküszöbölhető, ugyanis amíg az adatcsere az egyik ~ segítségével folytatódik, a program feldolgozhatja az előzőleg feltöltött ~ tartalmát v. új adatokkal töltheti fel a következőkben kiíratandó ~t (→puffertechnika).

puffermemória (buffer memory) – →puffer

pufferregiszter (buffer register) – speciális rendeltetésű →regiszter, amely két →hardveregység közötti átvitel közben az adatokat ideiglenesen tárolja. Ez különösen akkor szükséges, amikor a két egység műveleti sebessége között jelentős különbség mutatkozik.

puffertechnika (buffering) – a →b/k műveletekhez szükséges →pufferek kijelölésére, működésük ütemezésére és kényelmes elérésére szolgáló módszerek összessége.

1) Egyszerű pufferelésnél mind a beviteli, mind a kiviteli műveletek számára egy-egy külön puffert jelölnek ki. Ezek mérete megegyezik a →fizikai bejegyzések (blokkok) maximális hosszával. A pufferek feltöltése, ill. kiürítése idejére a felhasználói program végrehajtása felfüggesztődik. Ez a módszer bármilyen szervezésnél használható.

2) Váltó pufferelésnél a b/k műveletek számára több puffert foglalnak le, s azokat egymás után töltik fel, ill. ürítik ki. Ez a módszer lehetővé teszi, hogy a feldolgozás párhuzamosan haladjon az adatcserével, ugyanis amíg az egyik puffer feltöltése folyik, a másik tartalma feldolgozható. Csak →sorbaállító adatelérési technikával használható.

R

Radix-50 ábrázolás (Radix-50 notation) – →alfanumerikus karakterek különleges tárolási formája, amellyel 3 karaktert lehet elhelyezni egy tizenhat bites szóban. Megvalósítása a következő összefüggés alapján történik:

$$C_1 \cdot (50)_8 + C_2 \cdot (50)_8 + C_3$$

rajzgép (plotter) – olyan →kimeneti egység, amely a számítógép által szolgáltatott adatokat grafikus ábra (pl. függvénygörbe) formájában szolgáltatja. Az írófej mozgására vonatkozó információk bináris vektorok formájában érkeznek a számítógépből, s ezeket alakítja át a ~ vezérműve vezérlőjelekké. A korszerű ~ek több színnel is rajzolhatnak (a színattribútumok szintén bináris információ formájában érkeznek).

raszteres kijelző (raster display) – az →interaktív (számítógépes) grafika egyik fontos eszköze (→grafikus kijelző). Működése megegyezik a hagyományos tévékészülékek működésével: a képet pontok sokaságából állítja elő, és ezeket az elektronsugár az ernyőt végigpásztázva villantja fel, vagyis a képet sorokra, a sorokat pedig pontokra bontva jeleníti meg (→bittérképezés, →felbontás).

redundancia (redundancy) – általános értelmezésben: valamilyen cél érdekében több eszköz felhasználása, mint amennyire ténylegesen szükség volna. A számítástechnikában (is) elsősorban a rendszerek megbízhatóságának növelésére, a hibákkal szembeni tűrőképesség erősítésére használják. Jellegzetes ~alkalmazások pl. a →paritásbit, az addicionális karakterek, az ellenőrző összegek, a hasonló típusú funkcionális egységek párhuzamos használata (moduláris redundancia, →tartalék redundancia). Ezeket általában preventív ~nak is nevezik, s mint olyan, pozitív szerepet játszik a rendszerek hibamentes működésében.

A ~ lehet negatív hatású is, ha a többleteszközök feleslegesen vannak jelen s a rendszer általános hatékonyságát rontják. Ilyen helyzet állhat elő pl. a különböző információs rendszerekben, ha azonos adatok többszörösen is jelen vannak. Az →adatbázisok megjelenése és elterjedése éppen azt a célt szolgálja, hogy a ~t a minimálisra csökkentse anélkül, hogy az a rendszer megbízhatóságát veszélyeztetné. Ez a minimális és ellenőrzött ~ két szinten is jelentkezhet az adatbázisban: fizikai ~ áll fenn akkor, ha ugyanaz az adatelem több helyen is megjelenik a fizikai adatbázisban; logikai ~ áll fenn akkor, ha ugyanazokat az adatelemeket többször is leírják az adatbázis szerkezetét meghatározó →adatbázissémában.

redundáns kód (redundant code) – olyan →kód, amely több bitpozíciót használ, mint amennyire a rögzítendő információk ábrázolására ténylegesen szükség volna. Egyszerű hibák felismerésére szolgál.

regiszter (register) – szekvenciális v. párhuzamos elektronikus áramkör, amely egy (v. több) →bájt hosszúságú bitvektor tárolására alkalmas. Funkcióját tekintve lehet általános rendeltetésű munka~ v. egyedi funkciót betöltő speciális ~ (pl. →akkumulátor, →bázisregiszter, →indexregiszter, →utasításszámláló regiszter stb.).

reguláris nyelv (regular language) – →reguláris nyelvtan által generált programozási nyelv (→Chomsky-hierarchia).

reguláris nyelvtan (regular grammar) – a →Chomsky-hierarchia szerinti osztályozás alapján a negyedik csoportba tartozó →generatív nyelvtan. A →helyettesítési szabályokra vonatkozó megszorítás meghatározza az egy lépésben létrehozható változók és alapjelek számát. Amikor legfeljebb egy változó használható a szabály mindkét oldalán, a helyettesítést lineárisnak nevezik. Ha a változó a szabály jobb oldalának jobb végén helyezkedik el, a helyettesítési szabály jobb lineáris. Ha a változó a szabály jobb oldalának bal végén helyezkedik el, a helyettesítési szabályt bal lineárisnak nevezik.

rekord (record) – az →[adat]állományt alkotó adathalmaz egyedére vonatkozó olyan lineáris adatszerkezet, amely az egyed tulajdonságainak összességét hordozza. Más megközelítésben a ~ az [adat]állománynak a feldolgozás szempontjából legkisebb logikai egysége. Az adatfeldolgozás logikai, ill. fizikai szemléletének megfelelően a ~nak kétféle értelmezése ismeretes: →logikai rekord és →fizikai bejegyzés.

rekord [adat]típus (record [data] type) – →erősen típusos nyelvekben definiálható,

rögzített számú heterogén adat együttes kezelésére használható →strukturált [adat]típus. Értékkészletét az elemek értékkészletének hatványhalmaza adja, megengedett műveletei az értékadás és az összehasonlítás (egyenlő, nem egyenlő). Az adatstruktúrát felépítő elemek maguk is lehetnek összetett struktúrák. Az elemek szelektora nem számított érték, mint a →tömb [adat]típusnál, hanem rögzített mezőazonosító. A ~ egyszerűbb változata egyes klasszikus programozási nyelvekben (pl. →COBOL, →PL/1) is megtalálható.

rekurzív adatszerkezet (recursive data structure) – olyan elemekből felépülő összetett adatszerkezet, amelyeknek típusa ugyanaz lehet, mint az azt tartalmazó adatszerkezeté. Pl. a →fa típusú adatszerkezet egy eleme lehet ismét egy fa, de lehet elemi adat is.

rekurzív algoritmus (recursive algorithm) – rekurzív folyamatok, számítások leírására alkalmas olyan →algoritmus, amelyben az egész eljárás újraelvégzését aktivizáló lépés található (vagyis önmagát közvetlenül v. közvetve újrakezdheti). Egyes →programozási nyelvek (pl. az →ALGOL, a →PL/1, a →Pascal stb.) megengedik a ~ok használatát, →asszembler nyelven pedig mindig megvalósítható.

rekurzív eljáráshívás (recursive procedure call) – az →eljáráshívás speciális megvalósítása, amely lényegében azt jelenti, hogy az eljárás önmagát hívja, aktivizálja újra. A ~ közvetlen, ha az aktív eljárás önmagát hívja, és közvetett, ha az aktív eljárás más aktivizált programegységen keresztül hivatkozik önmagára.

rekurzivitás (recursion) – valamely objektum (jelenség, eljárás, program, rajz stb.) ama tulajdonsága, hogy leírása, elemzése, felépítése stb. hasonló típusú objektumok segítségével megvalósítható. Programozástechnikai szempontból a →felülről lefelé haladó módszer tulajdonsága, amelynek értelmében a kiszámítandó mennyiséget, a megszervezendő folyamatot sorra lebontják összetevőire. Így pl. n! kiszámításánál az n! = n · (n–1)! összefüggésből indulnak ki, majd (n – 1) = = (n – 1) · (n – 2)! következik, és így tovább, egészen addig, amíg 1! = 1.

relációfeltétel (relation condition) – egyes →programozási nyelvekben használható →egyszerű feltétel, amely egyszerű →programelágaztatást tesz lehetővé két operandus aktuális értékének nagyság szerinti összehasonlítása alapján. Numerikus operandusoknál az összehasonlítás azok algebrai értékére vonatkozik, vagyis előjelhelyes. Nemnumerikus operandusok összehasonlítása a karakterkészlet nagyság szerinti sorrendjén alapszik, amelynek meghatározása az egyes gépi implementációk hatáskörébe tartozik.

relációs adatbázis (relational data base) – olyan →adatbázis, amelyet E. F. Codd által javasolt technikával, vagyis a relációk felhasználásával állítottak elő.

A relációt egymáshoz hasonló egyedek bizonyos tulajdonságait leíró táblázatnak lehet tekinteni. A táblázat megnevezése a relációnév, minden sora egy n-es, minden oszlopa egy attribútum, és az oszlopban csak a megfelelő tartományból vett értékek fordulhatnak elő.

A relációs adatmodell műveleteket is definiál a relációkon.

1) A kiválasztás (szelekció) logikai kifejezést alkalmaz egy relációra, és kiválasztja belőle azokat az n-eseket, amelyekre a kifejezés igaz.

2) Az összekapcsolás két reláció n-eseiből alkotott párokból választja ki az előírt feltételeknek megfelelő párokat. A feltétel a két relációban előforduló közös tartományon értelmezett attribútumok értékére vonatkozik.

3) A vetítés (projekció) egy reláció attribútumainak egy részhalmazát „vágja ki" a reláció minden n-eséből, és az esetleg keletkező azonos sorok közül elhagyja a feleslegeseket.

4) A relációt n-esei halmazának tekintve, halmazműveleteket (egyesítés, metszet, különbség) lehet végrehajtani olyan relációk között, amelyeknek attribútumai páronként közös tartományon vannak értelmezve.

A felsorolt műveletek relációkon működnek és relációt adnak eredményül, ezért relációs algebrai műveleteknek is nevezik őket.

relációs kifejezés (relational expression) – →relációs operátorral összekapcsolt két aritmetikai kifejezés, amely kiértékelés után logikai értéket ad eredményül, így logikai operandusként vehet részt a →logikai kifejezésekben. Az összehasonlításban szereplő arit-

metikai kifejezések értelmezése változó a különböző →programozási nyelvekben. Pl. a →FORTRAN-ban csak algebrai kifejezések lehetnek, a →COBOL-ban és a →PL/1-ben viszont →jelsorozatok, ez utóbbiban ezenkívül bitsorozatok és →mutatók is szerepelhetnek.

relációs operátor (relational operator) – a →relációs kifejezések felépítésében szereplő olyan →lefoglalt szó, speciális karakter v. ezek kombinációja, amely viszonyítási műveletet fejez ki. Szintaktikai jelölésmódja változik a különböző magas szintű →programozási nyelvekben.

relatív cím (relative address) – valamely címzési alaphoz viszonyított hivatkozás a →gépi utasítás →címrészében. A viszonyítási alap lehet az illető utasításhoz tartozó →utasításszámláló regiszter tartalma (tkp. az utasítás aktuális címe) v. a →betöltési cím, ha a programot (programegységet) a program kezdetéhez viszonyítva címezték. Ha →bázisregiszteres címzést használnak, a címzési alapot zérónak tekintik, az utasításban pedig speciális mező van fenntartva annak a bázisregiszternek a kódolására, amelybe betöltés előtt a reális címzési alapot rögzítik. Ilyenformán a végrehajtás alatt az effektív cím a bázisregiszter tartalmának és a –nek az összegeként alakul ki. Ez a címzési mód a programok →áthelyezésének egyik megoldási lehetősége.

relatív címzés (relative addressing) – a →gépi utasítás →címrészének értelmezési módja, amikor azt a végrehajtás alatt lévő utasítás címéhez viszonyított pozitív v. negatív eltolásként v. pedig a programegység (→modul, →szegmens, →lap) kezdetéhez viszonyított eltolásként kezelik. A relatív címek leképzése →abszolút címekké a végrehajtás idejére halasztódik, s ez a legnagyobb előnye, ugyanis a csak –sel dolgozó programok az →operatív memóriába bárhova betölthetők, →áthelyezésük viszonylag könnyen végrehajtható.

relatív hiba (relative error) – olyan →hiba, amelynek mértéke az →abszolút hiba és a számított érték hányadosa.

rendelkezésre állás (availability) – a számítógéprendszerek teljesítőképességének egyik fontos jellemzője. Arra vonatkozik, hogy milyen mértékben működik üzembiztosan a rendszer akkor, amikor arra éppen szükség van. A teljesítőképesség másik két alapvető jellemzője az →áteresztőképesség és a →válaszidő.

rendellenes befejezés (abnormal termination) – valamely program végrehajtásának erőszakos megszakítása, amikor az →operációs rendszer olyan →hardver- v. →szoftverhibát észlel, amely lehetetlenné teszi a végrehajtás folytatását. Ilyen hiba lehet pl. nem létező →erőforrásra való hivatkozás, valamely hardveregység hibás működése v. teljes kiesése, nem kezelhető ismeretlen utasításkód, nem létező v. védett területen lévő címre való hivatkozás stb.

rendezés (sorting) – egy adathalmazhoz tartozó adatoknak a megfelelően kiválasztott szempont(ok) szerinti sorba állítása. Gyors –hez aránylag nagy →operatív memóriára, igen nagy mennyiségű adatnál viszont sok kiegészítő adathordozóra van szükség. Az adott probléma megoldására szolgáló optimális kompromisszumot a különböző –i módszerek közötti választás lehetősége biztosítja.

Az operatív memóriában megvalósított (belső) –nél, ahol az adatok közvetlenül címezhetők, a legismertebb módszerek a következők: →beszúrásos rendezés, →helycserés rendezés, →kieséses rendezés, →kiválasztásos rendezés, →kvadratikus rendezés, →összeválogatásos rendezés, →szétszórásos rendezés stb.

Ha nagyon nagy a rendezendő adatmennyiség, akkor ennek csak egy részét lehet az operatív memóriában feldolgozni. A beolvasott részhalmazt valamilyen módszer szerint rendezik, majd kiírják egy külső adathordozóra. A rendezett részhalmazokat (monotóniákat) később válogatják össze, hogy a végleges rendezett halmazt előállítsák (→összeválogatás).

rendezőprogram (sorting program) – a legtöbb →operációs rendszerben megtalálható →segédprogram; az adathalmazok (→táblázatok, →[adat]állományok) logikai elemeit olyan sorrendbe állítja, amelyet egy v. több rendezési kulcs növekvő v. csökkenő értékei határoznak meg. A rendezési kulcsnak a logikai elemekhez tartozó adatmezőnek kell lennie. A – gyakorlati megvalósításai rend-

szerint →programgenerátorok: a felhasználó által meghatározott konkrét rendezési feladatra →paraméterezett eljárásokból állítják össze a legmegfelelőbb ~ot.

rendszerállomány (system file) – speciális rendeltetésű →[adat]állomány, amelyet az →operációs rendszer a saját működése szempontjából fontos információk tárolására használ. Ilyen ~okban tárolják pl. a →fordítóprogramok a fordítási menetek során előállított közbenső kódokat. Minden több felhasználóval dolgozó operációs rendszernek van egy ~a, amelyben a számlázással kapcsolatos adatokat rögzítik. ~ként kezelik általában a →kötegelt feldolgozásnál a standard →bemeneti egységet, ill. a standard →kimeneti egységet is.

rendszerelemzés (system analysis) – számítógépes rendszerek megvalósításának első fázisa, amelynek során a rendszeren belül végbemenő folyamatokat nem egymástól elkülönülten, hanem a teljes rendszer szempontjából vizsgálják, és a vezérlést v. információszolgáltatást elsősorban az elektronikus számítógép felhasználása szempontjából közelítik meg. A ~t végző szakembereknek az új számítógépes rendszer bevezetése előtt a következő fő kérdésekben kell döntést hozniuk: Milyen folyamatok kerüljenek be a rendszerbe, kinek v. kiknek az igényeit elégítse ki elsősorban az új rendszer, mekkora és milyen költségek mellett gazdaságos még az új rendszer?

A ~ elsőrendű célja, hogy ellenőrizzék, vajon ezek a célok és követelmények reálisak-e, és ha igen, akkor biztosítsák megvalósításukat a számítógépeken. Meg kell állapítani, hogy az automatizálandó feladat hagyományos megoldásában használt információáramlás mennyire felel meg az új rendszer körülményei között, s végre kell hajtani a szükséges változtatásokat.

rendszereszköz (system device) – valamely →operációs rendszer nemrezidens részét tartalmazó tárolóeszköz. Rendszerint →mágneslemez v. →mágnesszalag.

rendszerfejlesztés (system design) – számítástechnikai vonatkozásban: valamilyen konkrét gazdasági, műszaki v. társadalmi folyamatban megvalósuló eljárás, valamint annak vezérlésére, ellenőrzésére, információáramlásának biztosítására szolgáló számítógépes rendszer megszervezése és megvalósítása. Ha a szervezés tárgya egy műszaki folyamat számítógépes irányítása, akkor →folyamatvezérlésről, ha a termelés rendje, akkor termelésirányításról, ha a vállalatok ügyviteli rendjének megszervezése, akkor ügyvitelszervezésről beszélnek. A ~ legfontosabb fázisai: a →rendszerelemzés, a rendszertervezés, a rendszer bevezetésének, alkalmazásának és fenntartásának megtervezése, a teljes rendszertechnikai folyamatra kiterjedő szervezés és irányítás.

rendszergenerálás (system generation) – az az eljárás, amelynek során előállítják egy adott →operációs rendszer konkrét →konfigurációra méretezett változatát. Kiindulásként az illető operációs rendszer alapváltozatát (az ún. →standard hivatkozási rendszert) használják, s a szabadon kiválasztható →hardver és →szoftver paraméterek megfelelő beállításával az adott számítógép konkrét konfigurációjának legjobban megfelelő operációs rendszert nyerik. Megvalósítására rendszerint speciális rendeltetésű rendszerprogram létezik, amely az illető operációs rendszer egy előbbi változatának felügyelete alatt működik.

A konfigurációk kiépítettsége tág határok között változhat (a →periferikus művek száma és típusa, az →operatív memória mérete, a →partíciók száma és mérete, a szoftver összetétele stb.), és ezt figyelembe kell venni a ~kor. A ~ bármikor megismételhető, ha ezt a hardverkonfiguráció bővítése v. az alkalmazási környezet változtatása szükségessé teszi.

rendszerkönyvtár (system library) – az →operációs rendszer számára előre meghatározott jellemzőkkel rendelkező, a →rendszereszközön található olyan →könyvtár, amelyben az operációs rendszer által használt programok, →szubrutinok találhatók. Itt rögzítik pl. →fordítóprogramok, →kapcsolóprogramok, →betöltőprogramok, →kezelőprogramok stb. közvetlenül végrehajtható formájú változatait, a →standard függvényeket megvalósító szubrutinokat stb.

rendszerlemez (system disk) – →operációs rendszert (is) tartalmazó →mágneslemez.

rendszer-makroutasítás (system macroinstruction) – olyan →makroutasítás, amelyet a programozónak nem kell definiálnia, mert a

megfelelő →makrodefiníciót előre beépítették az adott →fordítóprogram táblázataiba. ~okat szoktak általában használni az adatátvitelre vonatkozó utasítások tömör jelölésére, a →felügyelőprogram erőforrásainak igénylésére (→felügyelőprogram-hívó utasítás), az →[adat]állományok előállításával, leírásával és kezelésével kapcsolatos műveletek jelölésére stb.

rendszernapló (system log) – az →operációs rendszer által kezelt speciális →[adat]állomány (→rendszerállomány), amelybe a számítógéprendszer működésével kapcsolatos legfontosabb információkat rögzíti maga az operációs rendszer. Ezek az adatok szolgálnak a számítógéprendszer karbantartói számára esetleges vitás események utólagos ellenőrzésére, különböző rendszerelemek működésének követésére, bizonyos gazdasági mutatók számítására stb.

rendszerösszeomlás (system crash) – →operációs rendszerben fellépő →végzetes hiba utáni állapot, a rendszer teljes megbénulása. Rendszerint az operációs rendszer újragenerálását (→rendszergenerálás) v. →visszaállítási másolatról történő újrabetöltését követeli.

rendszerprogram (system program) – a felhasználói programfejlesztést hatékonyan segítő, az →operációs rendszer funkcióit kiegészítő program. Elsősorban azért hozzák létre, hogy a számítógépet felhasználói számára kihasználhatóbbá tegyék, valamint azért, hogy a felhasználói programok elkészítésének technikai feladataiban maga a gép nyújtson segítséget a programozónak. A legfontosabb ~ok a →fordítóprogramok, az →értelmezőprogramok, a →kapcsolóprogramok, a →betöltőprogramok, másolóprogramok stb.

rendszerprogramozó (system programmer) – →operációs rendszerekhez tartozó programtermékek tervezésében és előállításában dolgozó szakember.

rendszerszoftver (system software) – mindazoknak a programoknak az összefoglaló elnevezése, amelyekkel egy számítógéprendszert elláttak, amelyeken keresztül tehát a felhasználók magát a számítógéprendszert látják. Alapvető feladata a programozással kapcsolatos rutinmunkák elvégzése úgy, hogy a számítógéprendszer →erőforrásait a lehető legjobban ki lehessen használni. A ~ magában foglalja az →operációs rendszert, a →rendszerprogramokat, a →kezelőprogramokat stb.

rendszerváltozó (system variable) – →operációs rendszereknek a konkrét →konfigurációnak megfelelő értékeit tartalmazó változó. A →rendszergenerálás egyik legfontosabb feladata éppen a ~k megfelelő értékekkel való feltöltése. A ~k tartalma általában újabb rendszergenerálás v. esetleg →inicializálás útján változtatható.

rezsiművelet (housekeeping) – olyan művelet, amelyet a számítógép →központi vezérlőegysége, ill. az →operációs rendszer a saját →erőforrások belső kezelésére hajt végre. A ~ek időigénye adja éppen azt a többletidőt, amelyet →többletráfordítás néven tartanak számon.

rögzített hosszúságú rekord (fixed length record) – olyan →rekord, amelynek hossza megegyezik az ugyanahhoz az →[adat]állományhoz tartozó összes többi rekord hosszával (más szóval: minden rekord ugyanolyan hosszúságú). Amennyiben a ~ok blokkoltak (→blokkolás), az egyes →fizikai bejegyzésekben tárolt ~ok száma is rögzített, s így az egyes blokkok hossza is állandó.

rögzített partíciók módszere (fixed-partitions method) – a →particionált memóriakezelés egyik fajta megvalósítása. Az →operatív memóriát (tkp. →címtartományát) a →rendszergeneráláskor előre rögzített számú és hosszúságú partíciókra osztják (→gépkezelői beavatkozással ezek a paraméterek változtathatók, de a felosztás természete megmarad). Az egyes partíciókban →kötegelt feldolgozás folyik. Meglehetősen merev módszer, bármilyen változtatás emberi beavatkozást igényel, az előzetes felosztás pedig nem felelhet meg a mindenkori igényeknek. Könnyen megtörténhet, hogy az egyes partíciók nincsenek teljesen lefoglalva, de a szabad hely kihasználatlan marad. A fejlődés természetes iránya a →változó partíciók módszere felé vezetett, ahol a felosztás dinamikus és az →operációs rendszer hatáskörébe tartozik.

RPG (Report Program Generator) – a hatvanas évek közepén kifejlesztett →programo-

zási nyelv, főleg kereskedelmi-ügyviteli területen használták a →COBOL nyelv kiegészítéseként. Elsősorban →[adat]állományok tartalmának olyan táblázatszerű megjelenítésére (nyomtatására) alkalmas, ahogyan azt az adatfeldolgozás útján nyert adatokat a felhasználói környezet megköveteli (táblázatok, jelentések, prognózisok stb.). Nagyon sok módszere később beépült magába a COBOL nyelvbe (→táblázatgenerátor), ill. az →adatbázis-kezelő rendszerbe.

rugalmasság (flexibility) – számítástechnikai értelmezésben a programtermékek változtathatósága, alkalmazkodóképessége mind a külső körülményekhez viszonyítva, mind az →operációs rendszer változásaival szemben. Szűkebb értelemben az operációs rendszerek kiépítettségi szintjére, a rendelkezésre álló →rendszerprogramok számára és minőségére utaló jellemző; a felhasználói programfejlesztést segítő, könnyebbé tevő eszközök és módszerek minőségét határozza meg.

S

sáv (track) – 1. →Mágnesszalag v. →lyukszalag hosszirányú zónája, amely a tárolt karakterek azonos bitpozícióit tartalmazza (mágneses állapot v. perforációk formájában). – 2. Közvetlen hozzáférésű mágneses adathordozókon (→mágneslemezeken) található koncentrikus kör alakú pályák egyike, amelyet fixfejes kivitelezésben külön író-olvasó fej, mozgófejes kivitelezésben pedig ugyanaz az író-olvasó fej sugárirányú elmozdulással szolgál ki. Az adatokat a fej helyzetének megváltoztatása nélkül lehet kiolvasni, ill. rögzíteni.

sávindex (track index) – →indexszekvenciális [adat]állománynál a legalsó indexszint neve (a közvetlenül felette található szint a →cilinderindex). A ~ rendszerint minden →cilinder nullás →sávján van. A cilinder minden egyes elsődleges adatterülethez tartozó sávjához tartalmaz egy bejegyzést, amely megadja az azon a sávon található legmagasabb értékű →kulcsot. A ~ tartalmazza még a túlcsordulási területen lévő, logikailag ehhez a sávhoz tartozó legmagasabb kulcsú rekord kulcsát és címét, az első túlcsordult rekord címét, valamint a túlcsordulási terület első szabad blokkjának címét.

sávszélesség (bandwith) – azon frekvenciák intervalluma, amelyeken egy →kommunikációs csatorna adatátvitelt képes lebonyolítani torzítás v. jelvesztés nélkül. A ~ mértéke tkp. a felső és alsó frekvenciahatárok különbsége. A ~ alapján a kommunikációs csatornák osztályozhatók: 1) keskenysávú csatornák (300 Hz alatt); 2) hangfrekvenciás csatornák (300-3000 Hz); 3) szélessávú csatornák (3000 Hz felett).

SD (Single Density) – egyszerű sűrűségű tárolást biztosító →hajlékonylemez típusjele (→tárolási sűrűség).

segédmemória (auxiliary memory) – →háttértár

segédprogram (service program) – →kezelőprogram

segélyállomány (help file) – →párbeszédes üzemmódban dolgozó programrendszerek kezelését megkönnyítő információkat tartalmazó speciális →[adat]állomány. Tartalmát szükség esetén egy-egy →funkcionális billentyű segítségével lehet megjeleníteni, rendszerint egy erre kijelölt →ablak területére. Lényegében a klasszikusnak számító →felhasználói kézikönyv szerepét játssza az →interaktív feldolgozás keretei között.

Sheffer-függvény (Sheffer's function) – kétargumentumú →logikai függvény, amely akkor és csakis akkor hamis, ha mind a két argumentum értéke egyszerre igaz. Igazságtáblázata:

	0	1
0	1	1
1	1	0

Shell-rendezés (Shell sort) – a →belső rendezés egyik módszere, tkp. a →helycserés rendezés változata. Lényege az, hogy a szomszédos elemek helyett az egymástól távol eső elemeket hasonlítják össze. Ilyen módon a nagymértékű rendezetlenség gyorsan csökken, az összehasonlított elemek távolsága egyre kisebb, amíg el nem éri az 1-et, s az eljárás tkp. átmegy a helycserés módszerbe.

SIMSCRIPT – diszkrét rendszerek →szimulációjára kifejlesztett magas szintű →programozási nyelv. A szimulált modellnek olyan program felel meg, amely tartalmazza a modell összes lehetséges állapotát az egyes események bekövetkezése után.

SIMULA – diszkrét rendszerek →szimulációjára kifejlesztett magas szintű →programozási nyelv, amely az →ALGOL 60 kiterjesztésének is tekinthető. Alapvető tulajdonsága az olyan folyamatok leírhatósága, amelyek – megfelelő felügyelet mellett – párhuzamosan is működhetnek. A diszkrét rendszer modelljét a dinamikus kölcsönhatásban lezajló folyamatok alkotják, az →időosztást speciális ütemezőprogram biztosítja.

SINIX – a Siemens cég által kifejlesztett →UNIX típusú →operációs rendszer.

sínrendszer (bus, highway) – valamely számítógéprendszer funkcionális elemei között kapcsolatot teremtő vezetékek összessége, amelyeknek segítségével bináris vektorokat alkotó jelek átvitele megvalósul. A bináris vektorok által kódolt információk természete alapján beszélnek adatsínről, címsínről, vezérlő sínről stb.

S-kifejezés (S-expression) – →egyirányú listák szöveges ábrázolásmódjának eszköze.

Elsősorban a listák feldolgozására kifejlesztett programozási nyelvek használják (→LISP).

SMALLTALK – a XEROX cég által kifejlesztett, az egyik legrégibb és legismertebb →objektumorientált nyelv.

SNOBOL – szimbolikus karakterfeldolgozásra kifejlesztett magas szintű →programozási nyelv, amelynek utasításai szimbolikus névvel ellátott →jelsorozatokkal végeznek műveleteket. Alapvető műveletei az értékadás, a láncképzés, az összehasonlítás, a helyettesítés, a részhalmazok kikeresése sorozatokból, a modellellenőrzés stb. A ~ legfontosabb felhasználási területe a természetes nyelvű szövegek elemzése, a →gépi fordítás stb.

sorbaállító adatelérési technika (queued access method) – alapvető →adatelérési technika, amelynek lényege az, hogy →logikai rekordokkal dolgozik, vagyis az →operációs rendszer adatkezelő szubrutinjai szinte minden adminisztratív szervezési műveletet (pl. blokkok tördelését, ill. felépítését stb.) automatikusan elvégeznek. Ugyancsak az operációs rendszer gondoskodik a →b/k műveletek és a feldolgozás szinkronizálásáról, vagyis a bevitelnél a kiürült →puffert azonnal feltölti (a programozó beavatkozása nélkül), s csak a tele puffert írja ki a perifériára (az utolsó, esetleg csonka blokkot az [adat]állományt lezáró utasítás küldi a perifériára). Tekintettel arra, hogy a rendszernek ismernie kell annak a blokknak a címét, ahonnan legközelebb olvasni, ill. ahová legközelebb írni kell, a ~ csak soros (szekvenciális) feldolgozáskor használható. Ennek megfelelően ismeretes szekvenciális és indexszekvenciális ~.

soros átvitel (serial transmission) – →bitsoros átvitel

soros elérésű memória (serial access memory) – olyan típusú →memória, amely bitsorozatok szekvenciális rögzítésére és átvitelére alkalmas. Jellegzetes képviselője a késleltetővonalas v. katódsugárcsöves memória. Alapvető jellemzői a kis memóriakapacitás és a nagy hozzáférési idő, ezért ma már nem használják.

soros hozzáférési módszer (serial access method) – →szekvenciális hozzáférési módszer

soros hozzáférésű adathordozó (serial access media) – olyan →adathordozó, amelyen a →fizikai bejegyzésekhez csak a felvitel sorrendjében (esetleg ellenkező sorrendben) lehet hozzáférni, s egy-egy bejegyzést fizikailag csak a sorban elfoglalt helye azonosít. A legfontosabb ~ok: a →lyukszalag, a →lyukkártya, a mágneskazetta, a →mágnesszalag stb.

soros keresés (serial search) – →lineáris keresés

soros távfeldolgozási mód (queued telecommunication access method) – →adatátviteli hálózatokban üzenetvezérlésre használt elérési módszer, amely széles körű lehetőséget biztosít távfeldolgozó vezérlőprogramok összeállítására adatgyűjtő, lekérdező, üzenetkapcsoló stb. feldolgozásokhoz. A megfelelő operációs rendszerek →makroutasítások segítségével lehetővé teszik az ilyenfajta programok megírását, ugyanis a makroutasítások révén kiválaszthatók és összekapcsolhatók azok a modulok, amelyekre a konkrét felhasználásban szükség van.

A ~ alapvető jelentősége abban áll, hogy a felhasználói program ugyanúgy használhatja a rendszer →termináljait, mint a számítógép klasszikus →perifériáját, vagyis a terminálról érkező, ill. az oda küldendő adatok éppen olyan makroutasításokkal kezelhetők, mint a közönséges soros írás, ill. olvasás. →Kötegelt feldolgozásnál a távfeldolgozási vonalon érkező →munka éppen úgy kerül ütemezésre, mint bármely „klasszikus" munka, és az →ütemező ezeket is kezelni tudja.

sorszámfüggvény (ordinal function) – →sorszámozott [adat]típust használó magas szintű →programozási nyelvekben (pl. →Pascal) olyan →standard függvény, amely meghatározza az →aktuális paraméterként megadott szimbólum sorszámát az [adat]típust meghatározó értékkészletben. Pl. ha a következő deklarációt használták:

var Nyelv: (magyar, német, angol)

akkor a ~: **ord**(német) értéke 1 lesz (a sorszámozás 0-val kezdődik).

sorszámozott [adat]típus (ordinal [data] type) – →erősen típusos nyelvekre jellemző minden olyan →[adat]típus, amelynek értékkészlete véges és rendezett halmaz, s így

egyértelmű kapcsolatba hozható a természetes számok egy részhalmazával. Alapvető jellemzője, hogy az értékkészlet bármely két eleméről bármikor eldönthető, melyik a nagyobb, ill. kisebb. Az erősen típusos nyelvek legismertebb —ai: az →egész, a →Boole-, a →felsorolt és az →intervallum [adat]típus. Bármely —on értelmezettek a →sorszámfüggvény, az →előző függvény és a →következő függvény. Megjegyzendő, hogy a →valós [adat]típus nem —.

speciális célú operációs rendszer (special-purpose operating system) – az →operációs rendszerek egyes funkcióit megvalósító olyan rendszer, amelyet rendszerint csak bizonyos feladatkör programjainak vezérlésére kell felkészíteni. A — ezért gyakran nem is különül el a feldolgozó programtól, hanem annak részeként jelenik meg. Ezekre a rendszerekre általában az jellemző, hogy a számítógépben egyetlen felhasználói program (tkp. alkalmazói rendszer) fut, de ez egyszerre több felhasználót szolgál ki párhuzamosan. Jellegzetes —ek: az interaktív rendszerek programok kifejlesztésére és futtatására egy adott nyelv keretein belül (pl. a BASIC rendszer); az interaktív rendszerek adatgyűjtési és folyamatvezérlési feladatok irányítására; az interaktív rendszerek információs lekérdezések vezérlésére stb.

SS (Single Side) – egyoldalas →hajlékonylemez típusjele.

standard függvény (standard function) – tudományos, műszaki, szimbolikus stb. problémák megoldására kifejlesztett magas szintű →programozási nyelvekben gyakran felhasznált matematikai v. speciális függvény elnevezése. A —ek általánosított →operátoroknak tekinthetők, ugyanis neveik rendszerint →kulcsszavai az illető nyelveknek. A —eket megvalósító eljárások közvetlenül végrehajtható formában vannak tárolva a →szubrutinkönyvtárban, s a →kapcsolóprogram automatikusan beépíti őket a hívóprogramba. Legismertebb —ek a logaritmusszámítások, trigonometriai függvények, négyzetgyökvonás, abszolútérték-számítás, konverziók stb.

standard hivatkozási rendszer (standard reference system) – valamely konkrét számítógéprendszer számára kifejlesztett alap- és felhasználói programok összessége. Az egyes konkrét felhasználók számára szükséges komponensek kiválasztását, ill. az azok működéséhez szükséges →szoftver, ill. →hardver paraméterek beállítását a →rendszergenerálás végzi el.

standard szubrutin (standard subroutine) – olyan egyedi funkciót ellátó programegység, amelyre általában minden, de rendszerint több felhasználói programnak is szüksége van. Ilyenek pl. a →b/k műveleteket végző, az →[adat]állományokhoz történő hozzáférést megvalósító stb. programegységek. A —okat általában →rendszerkönyvtárakban közvetlenül végrehajtható formában v. →áthelyezhető bináris formában tárolják, s a →kapcsolóprogram automatikusan beépíti őket a felhasználói programokba.

statikus adatstruktúra (static data structure) – olyan →adatstruktúra, amelynek szerkezeti felépítése időben változatlan (pl. →tömb). →Operatív memóriabeli ábrázolásához járulékos információk nem szükségesek, mivel a struktúrához tartozó elemek közötti viszony változatlan. (Különleges esetekben, mint pl. a hézagos mátrixoknál, használhatnak ugyan →láncolást, de ez nem változtatja meg a struktúra alapvetően statikus jellegét.) Ha az adatstruktúra szerkezeti felépítése változatlan, de az elemek száma időben változó (pl. →táblázat, →verem), esetleg →dinamikus memóriakiosztást támogató eljárásokra is szükség lehet.

statikus attribútum (static attribute) – valamely szintaktikai elem olyan tulajdonsága, amely már a →fordítás idején ismert, tehát az általa érintett →szemantikai szabályokat már a →fordítóprogram ellenőrizheti.

statikus kifejezés (static expression) – olyan kifejezés, amely már a program →fordítása alatt kiértékelhető. Modern programozási nyelvekben (pl. →CHILL, →Ada) van jelentősége az állandó értékek (→konstansok) deklarálásánál. Felépítésében →literálok, literálazonosítók és előre definiált operátorok vehetnek részt (a →változók jelenléte kizárt!).

statikus modell (static model) – adott időpontra vonatkozó →modell. Segítségével a modellezett jelenség egy időpontra vonatkoztatott alternatív lehetséges helyzeteit hasonlítják össze (szemben a →dinamikus modellel).

statikus szubrutin (static subroutine) – olyan ismételten végrehajtott programegység, amely a körülményektől függetlenül mindig ugyanaz(oka)t a funkció(ka)t látja el. Aktuális végrehajtásának vezérlésére tehát →paramétereket nem igényel (szemben a →dinamikus szubrutinnal).

statikus változó (static variable) – blokkszerkezetű magas szintű →programozási nyelvekben a →változók olyan típusa, amely a program teljes végrehajtása alatt létezik, s az azonosítóhoz rendelt attribútumok is változatlanok. Az →[adat]típusnak megfelelő hosszúságú és szerkezetű memóriazónát már a végrehajtás elején ténylegesen kiutalják, s ez a lefoglalás a végrehajtás végéig érvényes is marad (szemben az →automatikus és a →dinamikus változóval).

stratégiai játék (strategic game) – minden olyan →játék, amelyben az összes játékosnak – a fennálló játékszabályok keretei között – bizonyos stratégia alkalmazásával van befolyása a játék menetére. Ilyenek pl. a sakk, a gazdasági-üzleti játékok, a kártyajátékok bizonyos fajtái stb.

strukturált [adat]típus (structured [data] type) – →adatstruktúrák definiálására és kezelésére használt →[adat]típus. Programozástechnikai szempontból homogén v. heterogén típusú elemek csoportos kezelésére alkalmas, legfontosabb ismérvei az összetevők típusa, az elemek strukturálási, ill. adott elem kiemelési módja. Egyszerűbb változata a klasszikus nyelvekben is megtalálható: pl. a →tömb a →FORTRAN-ban, a →PL/1-ben stb., a →rekord a →COBOL-ban, a →PL/1-ben stb., de elméleti megalapozottsággal csak az →erősen típusos nyelvekben jelentkezett. A legismertebb ~ok: a →tömb [adat]típus, az →állomány [adat]típus, a →halmaz [adat]típus, a →rekord [adat]típus stb.

strukturált program (structured program) – a →strukturált programozás szabályai és módszertana alapján előállított program.

strukturált programozás (structured programming) – korszerű programfejlesztési módszertan, amelyet E. W. Dijkstra alapozott meg. Alapelve az, hogy a megvalósítandó programot modellsorozattal kell megközelíteni. Itt minden modellen olyan programot értenek, amely a célprogram működését valamilyen „magasabb intelligenciájú" gépen szimulálja. Természetesen ez a gép egy →absztrakt számítógép, amelynek megvalósítása nem szükségszerű, ugyanis a modellsorozat következő eleme az előzőnek egy konkrétabb kifejtése, egészen addig, amíg az utolsó változat egybe nem esik valamely létező gép (programozási nyelv) →utasításkészletével leírt programjával. A modellsorozat elemei tehát hierarchikus viszonyban állnak egymással, ez a hierarchia azonban minőségileg különbözik a →moduláris programozásból ismert hierarchiától. A ~ fontos alapelve a →lépésenkénti finomítás elve, amely arra ad módszert, hogyan alakítsák ki a meghatározott absztrakciós szinteket. Az egyes szinteken a programot jól áttekinthető, egyszerű vezérlőstruktúrák sorozataként adják meg. Ezek a struktúrák: az egyszerű utasíssorozat, a feltételtől függő döntésstruktúra és az iterációs szerkezet.

SZ

szabály alapú módszer (rule-based method) – →ismeretábrázolásban használható olyan módszer, amely „ha (feltételek), akkor (következmények)" alakú szabályokat alkalmaz. A „feltételek" rész tartalmazza mindazokat a kritériumokat, amelyeknek teljesülniük kell ahhoz, hogy az illető szabály alkalmazható legyen, vagyis a „következmények" rész megvalósulhasson.

szabványírás (standard writing) – adott formaelőírásoknak megfelelő, közvetlenül feldolgozható →bizonylatok kitöltésére használt írásmód.

szabványos címke (standard label) – →mágnesszalagon v. →mágneslemezen tárolt →[adat]állományokra, ill. magára az →adathordozóra vonatkozó azonosítási információk csoportosítása, amelyeknek feldolgozását az →operációs rendszer (→[adat]állomány-kezelő rendszer) biztosítja. Formája 80 →bájt hosszúságú blokk (→fizikai bejegyzés), amelynek első 4 karaktere a ~ azonosítója. A ~k három csoportba oszthatók: →adathordozói címkék, →[adat]állomány-kezdőcímkék és →[adat]állomány-zárócímkék.

szakasz (section) – →asszembler nyelven kódolt programok szerkezetileg önálló egysége, amelyet egymagában v. más ~okkal együtt lehet →tárgymodullá fordítani. A ~ban definiált azonosítók rendszerint →lokális szimbólumok, s csak az adott ~ban használhatók. Azokat az azonosítókat, amelyeket más ~oknak is kell használniuk, megfelelő utasításokkal →globális szimbólumoknak kell deklarálni. Kivételt jelent a csak leíró jellegű →álszakasz, amelynek összes azonosítója implicit módon globális szimbólum, s közösen használt memóriatartományt ír le.

szakértőrendszer (expert system) – a hetvenes években megjelent speciális rendeltetésű programtermék, amely sok tekintetben az emberi szakértőhöz hasonlóan old meg feladatokat. Az ember-gép kapcsolat megvalósításának sokat ígérő új megközelítése, különösen a →logikai programozás elméletének és módszertanának kidolgozása után. Lényegében ismeretek, gyakorlati tapasztalatok rendezett gyűjteménye, amelyeket emberi szakértők sokszor egy életen keresztül gyűjtöttek össze, és amelyeket igen ritkán lehet pontos matematikai formalizmussal leírni. Az emberi szakértőhöz hasonlóan a ~ is elsősorban szimbolikus következtetések és heurisztikus módszerek alkalmazásával jut el az eredményhez. Tkp. a számítógép teljesítőképességének és az emberi tapasztalatnak, feladatmegoldó képességnek szerencsés összekapcsolása, amely viszonylag szűk szakterületeken (pl. az orvostudomány egyes ágaiban, a molekuláris biológiában, a geológiában stb.) a tárolt →ismeretek jól irányított felhasználásával eredményesen és hatékonyan képes aránylag bonyolult feladatokat megoldani.

A ~ek egyik jellegzetes, már működő képviselője a Stanford Egyetemen kifejlesztett MYCIN rendszer, amely a vér fertőző betegségeinek felismerésére készült. A betegségek leírását (a rendszer „tudását") mintegy 450 szabály alkotja, a diagnózis módszertanának leírása pedig metaszabályokban és keresési stratégiákban valósul meg. A rendszer biztosítja az új ismeretek beillesztését is, és ellenőrzi a kialakult új tudáshalmaz ellentmondás-mentességét.

szalaglyukasztó (tape puncher/printer) – 1. Számítógéprendszer olyan →kimeneti egysége, amely az →operatív memóriában tárolt, binárisan kódolt információt átalakítja és azt a →lyukszalagra lyukasztja. – 2. Lyukszalag lyukasztására szolgáló, klaviatúrával ellátott berendezés, amelynek segítségével a felhasználók – a számítógép számára bemenő – információkat kódolhatnak. Ma már nem használják.

számítástechnika (computing technique) – mindazok az eszközök (gépek, készülékek, berendezések stb.) és az azokkal megvalósítható eljárások, módszerek, programok, amelyek számítási, vezérlési, általában információfeldolgozási feladatok automatizált megoldására alkalmasak. Bizonyos összefüggésekben a ~ fogalma egyre jobban eltolódik egy olyan, viszonylag új tudomány felé, amely a számítógépek, az automaták, az algoritmusok elméletét, a számítógépes nyelvészetet, a programozás módszertanát, a távközlés elméletét stb. foglalja magában.

számítógép (computer) – tágabb értelmezésben minden olyan berendezés, amely

megfelelő bemenő adatok alapján olyan kimenő adatokat képes előállítani, amelyek v. közvetlenül értelmezhetők a felhasználók részére v. más berendezések vezérlésére használhatók. Mai, szűkebb értelmezésben olyan elektronikus berendezés, amely információk (adatok + programok) tárolására alkalmas →memóriával rendelkezik, az adatok feldolgozásához →programra van szüksége, és saját tevékenységét, működését vezérelni képes.

Az első digitális ~nek tekinthető eszközt a nagy francia matematikus, Blaise Pascal készítette (az összeadás és kivonás elvégzésére alkalmas mechanikus szerkezet volt), amelyet a XVII. század végén W. Leibniz fejlesztett tovább (a szorzás és az osztás elvégzésére). A fejlődés következő fontos állomása Ch. Babbage 1823-ban elkészített táblázatszerkesztő „differenciagépe" volt. Ugyancsak Babbage nevéhez fűződik a ~ programozhatóságának felismerése, vagyis olyan gépek készíthetősége, amelyek aritmetikai és döntési műveletek hosszú sorát képesek elvégezni az ember beavatkozása nélkül. Babbage analitikai gépének gyakorlati megvalósítását a korabeli technológia nem tette lehetővé, csak 1939–44 között építette meg Howard Aiken professzor a MARK I ~et, amely végre megtestesítette Babbage elképzeléseit.

Az első elektronikus ~ az ENIAC (Electronic Numerical Integrator and Calculator) volt, amely a Pennsylvania Egyetemen készült el 1942–46 között. A mai értelemben vett ~ megvalósítása végül Neumann János nevéhez fűződik, akitől a programtároló (=→operatív memória) fogalma származik (→Neumann-gép). Legfontosabb újításával, a programok tárolásával lehetővé vált az utasítások adatként való kezelése, vagyis az, hogy a ~ módosítsa saját programját. Az első modern digitális ~, amelynek →architektúrája ma is irányt szabó a számítógép-tervezésben, az EDVAC (Electronic Discrete Variable Automatic Computer), 1952-re készült el.

számítógépcsalád (computer series) – a számítógépgyártó vállalatok rendszerint nem egy-egy számítógéptípust, hanem ezek egy-egy sorozatát bocsátják ki. Ezeknek az a jellemzőjük, hogy mind →hardver, mind →szoftver szempontból lentről felfelé kompatibilisek (vagyis a korábbi →konfigurációkon megvalósított programtermékek az újabb változatokon változatlanul v. minimális változtatásokkal végrehajthatók). A különbség elsősorban a teljesítményben és a kiépítettség mértékében nyilvánul meg (pl. abban, hogy mekkora az →operatív memória, milyen mennyiségű és minőségű →periféria építhető be a rendszerbe, mekkora a szoftverellátottság stb.).

számítógépes grafika (computer graphics) – mindazoknak a számítástechnikai eszközöknek és módszereknek az összessége, amelyek lehetővé teszik grafikus adatok bevitelét és kihozatalát, ill. feldolgozását úgy, hogy az ember és a számítógép közötti kapcsolat alapvetően grafikus jelleget mutasson.

Két alapvető irányzata különböztethető meg: a neminteraktív grafika, amelynél a képet digitális rajzgépek, mikrofilm-berendezések és más hasonló eszközök állítják elő, és az →interaktív grafika, amelynek eszköze elsősorban a →grafikus kijelző.

A ~ nagyon vonzó és hatékony megoldást ad számos területen, ugyanis a legtöbb esetben a vizuális információ felülmúlhatatlanul jobb az egyéb információmegjelenítési formáknál. Jellegénél fogva a műszaki tervezés a ~ egyik legnagyobb felhasználója. Bizonyos tervezési fázisokban a mérnök csaknem kizárólag rajzban gondolkozik, más fázisokban pedig a számítások eredményeit rendszerint grafikonokkal, eloszlásdiagramokkal, átviteli függvényekkel v. szerkezeteknek a terhelés hatására bekövetkezett deformációjával szemlélteti. Önálló felhasználási területet jelentenek a →végeselemek módszerét alkalmazó felhasználások.

A ~ egyik fontos felhasználása az elektromos áramkörök kapcsolási és huzalozási rajzainak tervezése. Ennek során a tervező előre megadott elemtípusokból állíthat elő áramköri vázlatokat. Bonyolultabb esetben az áramkörök matematikai modelljét is létre lehet hozni a kapcsolási rajzzal egy időben, és így mód nyílik az áramkörök viselkedésének számítógépes →szimulációjára, a szimuláció eredményeinek megjelenítésére.

Más elterjedt felhasználási területei a ~nak: a számítógéppel segített térképészet, az igazgatási és vezetési információs rendszerek, a →számítógéppel segített oktatás, a →számítógéppel segített tervezés, a →képfel-

dolgozás, az iparművészet és a szórakoztatóipar (pl. a →személyi számítógépeken megvalósított grafikus játékok, rajzfilmek) stb.

Módszereit tekintve két fő irányzata van: 1. vektorgrafika – koordinátákkal, vonalakkal, felületekkel dolgozik; 2. pixelgrafika – pontminták egymásra helyezésével működik.

számítógépes numerikus vezérlés (computerized numeric control) – a →numerikus vezérlés modern változata, amelynél a vezérlőadatokat számítógépes program állítja elő. Komoly fejlődést a →mikroprocesszorok megjelenése után mutatott.

számítógép-generáció (computer generation) – a számítógépek és rajtuk keresztül a számítástechnika fejlődésének minőségileg megkülönböztethető szakaszaira használt osztályozási kategória. A –t meghatározó fontosabb kritériumok: az →architektúrában felhasznált elektronikai komponensek minősége, a →memóriák tárolási kapacitása, a feldolgozási sebesség, a logikai szervezés, az →operációs rendszerek kiépítettségi foka, a →megbízhatóság, a teljesítmény-ár viszonya stb. Eddigi fejlődésük során a számítógépeknek négy generációja alakult ki, s az 1990-es évekre várható az ötödik megvalósítása.

Az eddig ismert négy generáció közös jellemzője a Neumann-elvre épülő architektúra, amelynek legfontosabb ismérvei: a →központi vezérlőegységet és az →operatív memóriát magában foglaló központi egység, az egyszintű →címtartomány, az alacsony szintű →gépi nyelv, a tárolt program által biztosított centralizált, szekvenciális vezérlés.

számítógép-hálózat (computer network) – olyan →elosztott rendszer, amelyben független számítógépek és számítógéprendszerek egymással információkat, adatokat cserélhetnek és →erőforrásaikat megosztva használhatják. Az információcserét adatátviteli eljárások (→protokollok) rendszere vezérli. Az egész rendszer igen rugalmas, ugyanis a feldolgozási kapacitást bármikor újabb feldolgozási egységek (számítógépek) bekapcsolásával növelni lehet. Több kisszámítógép –ba szervezésével jelentős gazdasági megtakarítást lehet megvalósítani, különösen ott, ahol biztonsági célokból bizonyos →tartalék redundanciát kell biztosítani. Valamely hiba csak egy részegység kiesését jelenti, egyszerű rekonfigurációval a megkezdett munka folytatható. A rendszerbe éppen nem foglalt számítógép helyi feldolgozásokra is felhasználható.

A – topológiai leírását olyan →gráffal lehet ábrázolni, amelyben a csomópontok a számítógépek, az élek pedig a →kommunikációs csatornák.

A legszélesebb körben használt –i formák közül:

1) a csillag – sugaras elrendezésű hálózat, amelyben bármilyen kommunikáció csak a központi számítógépen keresztül lehetséges, ezért ez utóbbinak megbízhatósága alapvetően befolyásolja az egész rendszer működését;

2) a rácsos – olyan hálózat, amelyben két csomópont között a kommunikáció több vonalon is lebonyolódhat, így működőképes marad esetleges vonalkieséseknél is; rendszerint egyenrangú feldolgozó központokat szerveznek ebbe a formába;

3) a hurkolt – olyan hálózat, amelyben a kommunikáció legfeljebb két vonalon bonyolódhat le;

4) a felfűzéses –ban a központot egyetlen vonal köti össze a legtávolabbi feldolgozó egységgel, a többi erről a vonalról ágazik le;

5) a vegyes –ban a fenti formák közül több is megtalálható.

számítógéppel segített gyártás (computer-aided manufacturing) – ipari tevékenységek számítógéppel támogatott területe. Igen széles skáláját jelenti a számítógépek alkalmazásának. Ide tartozik elsősorban a →szerszámgépek numerikus vezérlése, de igen hatékonyan kapcsolódik e területhez a →számítógéppel segített tervezés is. Sokan ehhez a területhez tartozónak tartják a raktárkészletek nyilvántartását, a technológiai ütemezést támogató programokat s újabban a robotok ipari alkalmazását is.

számítógéppel segített mérés és vezérlés (computer aided measurement and control) – a számítógépes alkalmazások egyik modern területe, elsősorban a →folyamatvezérlésben és az automatizált szállításban. A mérésadatok kiértékelésén alapuló, →zárt hatásláncú vezérlés problémáinak számítógépes megoldásával kapcsolatos eszközök és módszerek fejlesztésével foglalkozó alkalmazásterület.

számítógéppel segített oktatás (computer-aided instruction) – a számítástechnika felhasználásának egyik igen fontos, dinamikus fejlődést mutató területe. Az elsajátítandó ismeretanyag mennyiségi és minőségi színvonala egyre nehezebb helyzetbe hozza a klasszikus iskolarendszert, s csak a módszerek és eszközök minőségi változása oldhatja meg a felmerülő problémákat. A számítógépek felhasználása lehetővé teszi a tanulók számára, hogy a legkedvezőbb ütemben sajátítsák el az ismereteket, ellenőrizzék saját ismereteiket, előmenetelüket. A legkülönbözőbb területeken és a legváltozatosabb témák oktatására igen hatékony oktatórendszerek jelentek meg az óvodáktól az egyetemekig, a szakmunkásképzéstől az űrhajósok felkészítéséig. Ezek a rendszerek általában két szinten valósulnak meg. Külön alrendszer áll a tanárok rendelkezésére, akik az elsajátítandó ismeretanyagot programozzák, és külön, rendszerint →párbeszédes üzemmódban működő alrendszer a tanulók részére. Az oktatási rendszerek a kimondottan oktatói funkciók mellett olyan feladatokat is el tudnak látni, mint a tanulók vizsgáztatása, az előmeneteli adatok nyilvántartása és kiértékelése, a statisztikai kimutatások elkészítése stb. Ezek nagymértékben elősegítik az egész oktatás színvonalának emelését.

Minőségi előrelépést jelentett a ~ fejlődésében a →számítógépes grafika felhasználása. Ezzel az új eszközzel nemcsak új minőségi szintre emelhetik a vizuális tanulást és tanítást, hanem lehetővé válnak olyan szimulációk és kísérletek is, amelyek valós körülmények között igen költségesek v. egyenesen kivihetetlenek lennének (pl. járművezetés szimulálása, csillagászati kísérletek stb.).

A ~ egyik nem elhanyagolható előnye az is, hogy a tanulók szinte észrevétlenül megbarátkoznak a számítástechnikai eszközökkel, s ezeknek az ismereteknek nagy hasznát látják a későbbi gyakorlati tevékenységben is.

számítógéppel segített tervezés (computer-aided design) – a számítástechnika felhasználásának kiterjedt, fontos területe. A számítógép lehetővé teszi nagy mennyiségű, bonyolult számítások gyors és pontos elvégzését, a lehetséges termékváltozatok összehasonlítását és kiértékelését, a leendő létesítmények viselkedésének →szimuláció útján történő vizsgálatát, a gyártmány-, a kezelési és a karbantartási dokumentáció előállítását és naprakész tartását stb.

Minőségi szintemelkedést jelentett a ~ben is a →számítógépes grafika megjelenése és igen gyors fejlődése, elsősorban az építőiparban és a gépiparban, ahol a tervezésben és a kivitelezésben egyaránt igen sok rajzra van szükség.

számított cím (computed address) – az a →cím, amelyet a →címmódosítás lehetőségeit is figyelembe véve nyernek. Rendszerint az →eltolás és a →bázisregiszter és/vagy →indexregiszter tartalmának összege.

számjegy (digit) – 1. Különböző számrendszerekben használatos szimbólum, amely az illető rendszerben a számok helyértékes ábrázolására alkalmas. Mindig a →számrendszer alapjánál kisebb egész értéket képvisel. A számítástechnika szempontjából jelentős számrendszerekben a következő ~eket használják:

. bináris számrendszer: {0,1} – alapszám 2;

. oktális számrendszer:
{0,1,2,3,4,5,6,7} – alapszám 8;

. decimális számrendszer:
{0,1,2,3,4,5,6,7,8,9} – alapszám 10;

. hexadecimális számrendszer:
{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}
– alapszám 16.

2. →Programozási nyelvek →terminális szimbóluma.

számológép (calculator) – mechanikus, elektromechanikus v. elektronikus működésű, az aritmetikai műveletek elvégzésére alkalmas eszköz. Alapvető, a →számítógéptől megkülönböztető jellemzője az, hogy programozásra nem alkalmas, a számolás minden lépésénél emberi beavatkozást igényel.

számrendszer alapja (radix) – valamely számrendszerben a számok ábrázolására használt szimbólumok száma. Minden, az alapszámnál nagyobb szám egyértelműen előállítható az alapszám olyan pozitív és negatív kitevőjű polinomjaként, amelynek együtthatói éppen a felhasználható szimbólumok. A számítástechnika szempontjából különleges jelentőségű a bináris számrendszer

(alapszáma 2), az oktális számrendszer (alapszáma 8), a decimális számrendszer (alapszáma 10) és a hexadecimális számrendszer (alapszáma 16).

származtatott adatelem (derived data item) – olyan adatelem, amely nem szerepel explicit módon a →fizikai adatbázisban, de amelynek értéke különböző algoritmusok útján meghatározható az adatbázisban ténylegesen tárolt más adatelemek és a közöttük fennálló kapcsolatok alapján. Alapvető jellemzője, hogy bár nem szerepel fizikailag az adatbázisban, az →adatbázis-kezelő rendszer segítségével kezelhető (tehát érvényes leírás van róla az →adatbázissémában).

származtatott [adat]típus (derived [data] type) – az →Ada programozási nyelvben használható olyan új →[adat]típus, amelyet egy már létező [adat]típusból kiindulva lehet definiálni. A ~ értékkészlete és művelethalmaza megegyezik az eredeti típusével, mégis logikailag teljesen új [adat]típust képvisel, ennek összes következményével (→típusegyezés). Így a két típus közötti értékadás, ill. más műveletek tiltottak. A ~ örökli a kiinduló [adat]típus összes tulajdonságát, de definíciója újabb attribútumokat is előírhat (pl. korlátozhatja az eredeti típus értékkészletét).

szatellita számítógép (satellite computer) – →számítógép-hálózatokban a központi (rendszerint nagy teljesítményű) számítógéphez →on-line üzemben kapcsolt kisebb számítógép, amelyen egyszerűbb, különösebb →erőforrásokra igényt nem tartó feladatokat oldanak meg. A ~ autonóm módon is működhet, amikor helyi feldolgozást végez, de elláthatja a központi számítógépen megoldandó feladatok előkészítését, utólagos feldolgozását v. részfeladatok megoldását is.

szegmens (segment) – 1. →Átfedési szerkezettel rendelkező programoknak a →betöltés szempontjából vett szerkezeti egysége, amely betöltéskor az →operatív memória összefüggő területét foglalja el. Rendszerint a programozó előírásai alapján a →kapcsolóprogram alakítja ki egy v. több →tárgymodul összekapcsolásával; betöltéséről az →operációs rendszer v. a felhasználó gondoskodik. – 2. A →virtuális memória →szegmentálásos megoldásánál a nagy →logikai címtartomány egymással határos, esetleg változó hosszúságú szakasza. Minden logikai cím rendezett (s,w) párosnak tekintendő, ahol s a ~ címe (azonosítója), a w pedig a ~en belüli relatív cím.

szegmentálás (segmentation) – 1. →Átfedési szerkezet használatakor az az eljárás, amikor a programozó a saját programját olyan egységekre bontja, amelyeknek nem kell egyidejűleg az →operatív memóriában tartózkodniuk. Ezek az egységek a →szegmensek, s →betöltésükről a programozó v. a programozó utasításai alapján a →felügyelőprogram gondoskodik, ugyanis az átfedési szerkezetet már →forrásnyelvi szinten, de legkésőbb a →kapcsolás előtt definiálni kell. – 2. →Virtuális memória használatánál az az eljárás, amikor az →operációs rendszer bontja logikai egységekre a lineáris szerkezetként felfogott felhasználói programot. A logikai szegmenseket rendszerint speciális kapcsolóprogram állítja elő, amely mindegyiknek azonosító számot ad és felépíti a szegmenstáblázatot. Az operatív memóriában a logikai szegmenst szekvenciálisan v. lapozott formában tárolják. A programban található hivatkozásokat c=(s,w) kétkomponensű v. c=(s,p,w) háromkomponensű logikai címekként kezelik, fizikai leképzésükről pedig az operációs rendszer gondoskodik (→címátalakítás).

szektor (sector) – a →mágneslemez legkisebb címezhető egysége. A →sávok több egyenlő (általában a 2 valamilyen hatványával egyenlő számú) részre vannak felosztva, s minden ilyen rész címezhető. A legtöbb rendszer megengedi, hogy a →fizikai bejegyzés egy ~nál hosszabb legyen, de mindig új ~on kell kezdődnie. Az →operatív memória és a →periféria közötti adatcsere egysége mindig a ~ egész számú többszöröse.

szekvenciális [adat]állomány (sequential file) – olyan →logikai [adat]állomány, amelyben az egymást követő →rekordok között egyértelmű logikai sorrend állapítható meg: adott rekord kulcsa nagyobb (v. kisebb) minden őt megelőző, és kisebb (v. nagyobb) minden őt követő rekord kulcsánál. A ~t az →adathordozó összefüggő zónájában rögzítik, s a rekordok csak szekvenciális sorrendben dolgozhatók fel. A ~ létrehozása előtt a rekordokat a szekvencialitást meghatározó kulcs szerint rendezni kell.

A ~ bármilyen típusú adathordozón rögzíthető. Nem címezhető adathordozón a rekordok fizikai elhelyezését is a meghatározott sorrendben kell elvégezni. Címezhető adathordozón (pl. →mágneslemezen) a rekordok rögzíthetők logikailag szekvenciálisan is, ahol az előírt sorrendet →láncolással valósítják meg. A ~t logikai szinten az →[adat]állománykezelő rendszer specializált szubrutinjai dolgozzák fel, s ezekre hárul a →blokkolással és az esetleges váltópuffereléssel (→puffertechnika) kapcsolatos adminisztratív tevékenység is.

szekvenciális alap-adatelérési technika (basic sequential access method) – →szekvenciális (szervezésű logikai) [adat]állományok feldolgozására alkalmazott →alapadatelérési technika.

szekvenciális algoritmus (sequential algorithm) – olyan algoritmus, amelyben a résztevékenységek egymást követve, nem egymással párhuzamosan hajtódnak végre (mint a →parallel algoritmusnál). Az egyetlen →processzorral rendelkező számítógépek csak ~okat hajthatnak végre. A →multiprogramozást megvalósító rendszerekben is csak látszólagos a párhuzamos végrehajtás, hiszen az egyetlen →központi vezérlőegység adott pillanatban csak egy résztevékenység végrehajtásával foglalkozhat. Programozástechnikai szempontból bizonyított, hogy bármely ~ három alapvető →vezérlőszerkezet segítségével leírható.

szekvenciális gép (sequential machine) – szekvenciális szinkron áramkörök tanulmányozására alkalmas matematikai modell (tkp. →determinisztikus véges automata).

szekvenciális hozzáférési módszer (sequential access method) – bármilyen típusú →adathordozón használható elérési módszer. Lényege az, hogy a tárolt adatok csakis a rögzítés sorrendjében nyerhetők vissza. Nem címezhető adathordozón (pl. →mágnesszalagon) a rögzítés mindig fizikailag szekvenciális, így az adatok visszanyerése is csak az adathordozón elfoglalt fizikai sorrendben történhet. Címezhető adathordozón (pl. →mágneslemezen) a szekvencialitás lehet logikai is, a →rekordok sorrendiségét →láncolással biztosíthatják.

szekvenciális sorbaállító adatelérési technika (queued sequential access method) – →szekvenciális (szervezésű logikai) [adat]állományok feldolgozására alkalmazott →sorbaállító adatelérési technika.

szelektív [adat]állomány (selective file) – →direkt szervezésű [adat]állomány

szelektív hozzáférés (selective access) – olyan adatok rögzítési, ill. visszanyerési módja, amelyek között nem lehet logikai sorrendet felállítani. Az ilyen adatokat →direkt szervezésű [adat]állományokként szokták kezelni, amelyekben az egyes →logikai rekordok valamely tulajdonsága és a címezhető →adathordozón elfoglalt helye között egyértelmű (v. majdnem egyértelmű) összefüggést határoznak meg.

szelektorcsatorna (selector channel) – olyan →adatcsatorna, amely egyetlen be-, ill. kiviteli műveletet tud a →központi vezérlőegységgel egy időben végrehajtani. Ez rendszerint egy v. több blokk átvitelét jelenti. Ez idő alatt a ~t egyetlen →periferikus egység teljesen lefoglalja: más egységek rajta keresztül csak akkor kommunikálhatnak, ha a már megkezdett átvitel befejeződött. Nagy sebességű adatátvitelt tesz lehetővé, azonban lassú periferikus eszközöknél gazdaságtalan: bár több készülék is kapcsolódhat ugyanahhoz a ~hoz, egyszerre csak egy lehet aktív, mivel a ~ egyszerre csak egy készülék adatátvitelét tudja ellátni. Ha egynél több készülék csatlakozik, akkor megfelelő elsőbbségi rendszer biztosítja a kívánt sorrend szerinti kiszolgálást. Rendszerint nagy sebességű perifériák vezérlésére használják, mágnesszalagoknál, mágneslemezeknél stb.

szemafor (semaphore) – párhuzamosan futó, konkurens folyamatok vezérlésére használatos olyan speciális célú, egész értékű változó, amelyet a folyamatok egy várakozó sorához kölcsönösen és egyértelműen hozzárendelnek. Az s ~on a folyamatok két primitív, egymást kizáró utasítást hajthatnak végre:

P(s), amelynek hatása a következő műveletsor:

1. s: = s – 1;
2. ha s < 0, akkor az utasítást végrehajtó folyamat futása megszakad, és a folyamat beáll az s ~hoz rendelt várakozó sorba;

V(s), amelynek hatása a következő műveletsor:

1. s: = s+1;
2. ha s < 0, akkor az s –hoz rendelt várakozó sorból egy folyamat kilép, és folytatja futását.

A P(s) és V(s) primitív utasításokat úgy szokták megvalósítani, hogy azok a 2. pontban megadott feltételek teljesülésénél →megszakítást eredményeznek, és az →operációs rendszer kapja meg a vezérlést. A P(s) utasítás után az operációs rendszer gondoskodik a megszakított folyamat állapotának megőrzéséről, és rögzíti annak jellemző adatait az s –hoz rendelt várakozási sor adatai között. A V(s) utasításnál viszont megfelelő kiválasztási algoritmus alapján meghatározza a várakozási sorból kilépő folyamatot, és gondoskodik annak folytatási feltételeiről.

szemantika (semantics) – 1. Általános értelmezésben az a tudomány, ill. szemiotikai és nyelvészeti tudományág, amely a szimbólumok és a hozzájuk rendelt értelmezés közötti kapcsolatokat tanulmányozza. – 2. Számítástechnikai értelmezésben a – adja meg a programok jelentését. Egy programozási nyelv definíciója csak akkor teljes, ha annak →szintaxisát és –ját egyaránt meghatározzák. A – megadása legegyszerűbben valamely természetes nyelven kifejezett szabályokkal lehetséges. Másik lehetséges módja maga az illető nyelv →fordítóprogramja: ha ismert a generált →tárgyprogram –ja, a →jelentésmegmaradás elve alapján adottnak tekinthető a forrásnyelvű program –ja is. Bár a fordítóprogram elvileg pontosan definiálja az általa lefordított programozási nyelvet, szigorú értelemben nem lehet nyelvdefiniáló eszköznek tekinteni.

Tudományosan megalapozottabb módszer a programok szintaxisának és –jának együttes definiálására a →formális nyelvek használata: a program –ját annak az átalakításnak a leírásával lehet megadni, amelyet a program bemenő adatain végrehajt. Az első ilyen nyelv, amellyel a →PL/1 programozási nyelvet definiálták, a →bécsi definíciós nyelv.

szemantikai analízis (semantic analysis) – a számítástechnikában a →fordítás folyamatának egyik fontos fázisa, amelynek során a különböző programelemekhez tartozó szemantikai attribútumokat összegyűjtik, elemzik és feldolgozzák. A – során derülhet fény olyan hibákra, amelyek helyes szintaktikai formánál is megjelenhetnek: pl. →[adat]típusok deklarációja és felhasználása közötti konfliktusok, ugyanannak az azonosítónak többszörös deklarációja stb. A helyes szemantikai elemek alapján hajthatók végre a fordítás következő fázisai, a →memóriakiosztás és a →kódgenerálás.

szemantikai hiba (semantic error) – programozásban elkövethető hibatípus, amely a választott programozási nyelv felületes v. „egyéni" értelmezésének következménye. A –kat a fordítóprogramok sokszor nem képesek kiszűrni, ezért csak a programok rendszeres →tesztelésével küszöbölhetők ki.

szemantikai kiterjesztés (semantic extension) – valamely programozási nyelv gazdagításának módszere. Lényege a nyelvben már szereplő utasítások, ill. operátorok →szemantikájának kibővítése. Pl. egy programozási nyelv –ét jelenti, hogyha a + operátort alkalmassá teszik adatelemeken kívül adatstruktúrák elfogadására (így pl. →tömbökösszeadására) is.

szemantikai szabály (semantic rule) – →programozási nyelvekben használt szabály, amelynek segítségével az egyes szintaktikai elemekhez, ill. azok kölcsönhatásához jelentést, értelmezést, tulajdonságokat lehet kapcsolni. A klasszikus programozási nyelvekben a –ok legnagyobb részét implicit módon a →fordítóprogram foglalja magában, a modern nyelvekben (pl. →CHILL, →Ada stb.) már explicit programrész áll a programozó rendelkezésére a – előírására. A statikus –okat már a fordítás alatt elemezni lehet, megsértésüket a fordítóprogram jelezheti. A dinamikus –ok már nem ellenőrizhetők fordítás alatt, a fordítóprogram esetleg csak a végrehajtás idejére építhet a →tárgyprogramba megfelelő ellenőrző eszközöket. A dinamikus –ok megsértése mindig végrehajtási hibához vezet.

személyi számítógép (personal computer) – →mikroprocesszoros kiépítettségű, kisméretű, „tartós fogyasztási cikknek" szánt számítógép, a számítástechnikának az oktatásba, a szórakoztatásba, a háztartásba bevonult terméke. Központjában mikroprocesszor áll, →memóriája kizárólag félvezetős típusú. Az egyszerű otthoni –ek (home com-

puter) memóriakapacitása 16-64 Kbájt, a magasabb szintet képviselő professzionális ~eké (professional computer) 640 Kbájt, sőt több Mbájtig is terjedhet. →Háttértárként eleinte kizárólag kazettás magnetofont használták, ma már igen elterjedt a →hajlékonylemez és a nagy kapacitású →merevlemez. Megjelenítő eszközként az alacsonyabb kategóriák a közönséges tévékészüléket, a professzionális gépek képernyős (sokszor színes ábrázolásra is alkalmas) →megjelenítőt használnak. A ~ekhez →nyomtató is csatlakoztatható, kezdve a legolcsóbb hőnyomtatóktól egészen a lézernyomtatókig.

Programozását eleinte →gépi nyelven végezték, de hamarosan lehetővé vált az →asszembler nyelvű, majd →BASIC programozás is; ma már olyan magas szintű programozási nyelvek is használhatók erre a célra, mint a →C, a →Pascal, a →FORTH, sőt a →Prolog is.

A professzionális kategóriájú gépek fejlett →operációs rendszereket is tartalmaznak (pl. →CP/M, →MS-DOS, →UNIX, →XENIX), sőt →adatátviteli szoftverjük útján hálózatba is integrálhatók.

szerencsejáték (game of hazard) – minden olyan →játék, amelyben a játékosoknak – a fennálló játékszabályok értelmében – nem lehet befolyásuk a játék kimenetelére (pl. a kockajátékok, a kártyajátékok bizonyos fajtái). Matematikailag nem a →játékelmélet, hanem a valószínűségszámítás jellegzetes tárgya.

szerkesztés (editing) – 1. →Forrásnyelvű programok szövegszerű összeállítását, javítását, kiegészítését, fordításra való előkészítését magában foglaló művelet. Elsősorban →párbeszédes üzemmódban dolgozó rendszereknél használják, ahol speciális segédprogramok állnak a felhasználók rendelkezésére (→programszerkesztő). – 2. Eredményadatok kivitelre való előkészítése. Ezeket rendszerint már az →operatív memóriában olyan sorrendbe és formára hozzák, amelyek megfelelnek a →nyomtatón, ill. a →megjelenítőn elérni kívánt alaknak. A magas szintű programozási nyelvek a nyomtatási formák leírása (→nyomtatási sablonok, formátumok) alapján rendszerint automatikusan végzik el a kívánt átalakításokat. – 3. Szöveg rögzítése v. módosítása szövegszerkesztővel.

szerkesztési karakter (editing character) – magas szintű programozási nyelvekben használatos olyan szimbólum, amelynek segítségével meghatározható a kivitelre kerülő adatokon végzendő alaki szerkesztés (pl. a numerikus értékekre vonatkozó nyomtatási mező hossza, a tényleges tizedespont helyzete, az előjel nyomtatása, a vezetőnullák elnyomása, a táblázatok nyomtatása, a rovatok kialakítása stb.).

szerkezet nélküli [adat]állomány (undefined file) – →[adat]állományként kezelt olyan adathalmaz, amelyben semmiféle logikai kapcsolat nem létezik az adatelemek között (sem az adatelemek azonosítója és az [adat]állományban elfoglalt helye, sem a fizikailag egymás után következő adatelemek között). Folyamatosan követik egymást az adathordozókon, s ebből következik, hogy feldolgozásuk is csak sorosan, elhelyezkedésük sorrendjében valósítható meg. Csak fizikai szintű feldolgozásra ad módot, ezért erősen géporientált. Az →[adat]állomány-kezelő rendszer csak a →fizikai bejegyzések mozgatását végzi, minden más művelet a programozó feladata.

A ~t általában akkor használják, amikor nagy adattömegeket kell biztosítani, ill. biztonsági okokból tárolni, de feldolgozásukra csak ritkán kerül sor. Ilyenkor a gyors és egyszerű előállítás részben kiegyenlíti a viszonylagos hátrányokat.

szerszámgépek numerikus vezérlése (numerical control of tools) – gyártási folyamatok automatizálásának egyik igen fontos területe. Lényege az, hogy a szerszámgépet →lyukszalagon v. →lyukkártyán tárolt numerikus adatok felismerésére alkalmas bemenettel látják el; itt ezek olyan vezérlőjelekké alakulnak át, amelyeket a gép vezérlőegysége közvetlenül fel tud használni az előírt műveleteket végrehajtó alegységek vezérlésére. Komplexebb →numerikus vezérlésnél az elvégzendő műveletek kódjai közvetlenül a vezérlő számítógépből származnak (a szerszámgép és a számítógép →on-line rendszert alkot).

A vezérlőprogramok előállítására speciális programozási nyelveket fejlesztettek ki (pl. →APT, EXAPT stb.), amelyek lehetővé teszik a felületek geometriai leírását, a pontosság

meghatározását, a megmunkálás során bejárandó út leírását stb.

szessziós rétegprotokoll (session layer protocol) – →hétréteges referenciamodell

szétszórásos rendezés (distribution sorting) – a →rendezésnek az →operatív memóriában megvalósítható módszere, amelynek lényege a következő: a rendezendő halmaz elemeinek kulcsaiból valamilyen algoritmus alapján közvetlenül meghatározzák az illető elemeknek a rendezett halmazban elfoglalt helyét. Az eljárás nagy hátránya az, hogy rendszerint igen nehéz olyan algoritmust találni, amelyik megfelelő kis területre egyértelműen sorrendbe tudja rendezni a nagy tartományban szétszórtan rendezetlenül elhelyezkedő kulcsokat.

szimbolikus címzés (symbolic addressing) – →asszembler nyelvekre jellemző címzési módszer. Lényege, hogy a programok kódolásakor az egyes programzónákra (utasításokra, adatokra) →címkék, szimbolikus nevek segítségével hivatkoznak az illető rekeszek tényleges címe helyett. A programok →fordításakor minden egyes szimbolikus címhez →relatív címet v. →abszolút címet kell hozzárendelni, a hivatkozási helyekre pedig a megfelelő tényleges címet kell behelyettesíteni. Nagymértékben megkönnyíti a programozó feladatát, ugyanis felhasználhatja a →gépi nyelv nyújtotta összes lehetőséget anélkül, hogy figyelmet kellene fordítania az egyes utasítások, ill. az adatok által lefoglalt memóriára, mert ez az egész tevékenység az →asszemblerre hárul.

A magas szintű programozási nyelvek vonatkozásában ~nek lehet tekinteni minden →azonosítót, ill. címkét, ugyanis ezek a fordítás nyomán végeredményben szintén operatív memóriabeli címeket fognak azonosítani.

szimbolikus hivatkozás (symbolic reference) – →külső hivatkozás

szimbolikus utasítás (symbolic instruction) – →asszembler nyelvek utasításainak elnevezése. Minden egyes →gépi utasításnak megfelel egy asszembler nyelvű utasítás, és fordítva, minden egyes ~nak egy és csakis egy gépi utasítás felel meg. Az utasításkód a könnyebb megjegyezhetőség kedvéért a legtöbbször emlékeztető rövidítés (mnemonikus kód), az operandusokat →szimbolikus címek képviselik. Ezekkel a lehetőségekkel a programozó logikailag ugyanúgy alakíthatja ki programját, mintha az gépi nyelven készülne, azzal a különbséggel, hogy a címeket és a műveleti kódokat is szimbolikus jelkombinációkkal jelölheti.

szimbólum (symbol) – 1. Jel v. jelsorozat, amellyel valamilyen objektumot, állapotot v. fogalmat ábrázolnak, s amely az ember v. a számítógéprendszer által értelmezhető. Információelméleti, ill. adatfeldolgozási szempontból a fizikailag rögzített, feldolgozható, ill. átvihető ~ot →adatnak, az adathoz rendelt jelentést pedig →információnak nevezik. A →programozási nyelvek vonatkozásában különleges jelentősége van a →terminális szimbólumnak és a →nemterminális szimbólumnak, amelyeket a →generatív nyelvtanokban használnak. – 2. A →mesterséges intelligencia szempontjából ~ minden olyan karaktersorozat, amely a valós világ egy-egy konkrét fogalmát képviseli. A →szakértőrendszerek pl. kevésbé dolgoznak pontos algoritmusokkal, helyettük többnyire ~okat választanak a feladat fogalmainak ábrázolására, és ezeken végeznek heurisztikus műveleteket. Pl. a következő ~:

oldódás ((NaCl, H_2O), jó)

valószínű értelmezése: „a konyhasó jól oldódik vízben".

szimbólumfeldolgozó nyelv (symbol manipulation language) – szöveges (jelsorozatos) információk feldolgozására szolgáló eljárások programozását megvalósító magas szintű →programozási nyelv. A számításorientációjú nyelvekkel (→ALGOL, →FORTRAN stb.) szemben a ~ utasításai szimbólumokkal (betűkkel, jelekkel stb.) és ezek különböző struktúráival (→listákkal, →jelsorozatokkal stb.) végeznek műveleteket.

Egyes esetekben – különösen nyelvészeti orientációjú ~eknél – kötött formájú program rendezett utasítássorozata helyett szabályok rendezetlen v. részben rendezett halmazát használják fel a végrehajtandó műveletek meghatározására. Természetesen ekkor is szükség van egy „metaalgoritmusra", amely a szabályokban foglalt információk értelmezésére és a nekik megfelelő műveletek elvégzésére vonatkozik.

A ~eket főleg gondolkozási folyamatok modellezésére, matematikai tételek bizonyítására, logikai feladatok megoldására, nyelvek formális szerkezetének és logikai apparátusának vizsgálatára, információk sokrétűen kapcsolódó rendszerének tárolására és változó szempontok szerinti visszakeresésére stb. használják. A legismertebb ~ a →SNOBOL, a →LISP, a CONVERT, a COGENT stb.

szimbólumtáblázat (symbol dictionary) – →fordítóprogramok által használt lineáris adatszerkezet, amelyben a forrásprogramban megjelenő objektumokat (adatnevek, [adat]állománynevek, konstansok stb.) és azok attribútumait tárolják. Előállítása a →szintaktikai analízis feladata (→fordítás).

szimmetrikus lista (symmetric list) – →kétirányú lista

szimmetrikus mátrix (symmetric matrix) – olyan valós, kvadratikus →mátrix, amely megegyezik saját transzponáltjával; az elemekre fennáll a következő egyenlőség: $a_{ij} = a_{ji}$. Sok numerikus módszer könnyen megfogalmazható és végrehajtható ~ok segítségével.

szimplex csatorna (simplex channel) – távolsági adatátvitelben olykor használatos olyan →kommunikációs csatorna, amelyen egyirányú információáramlás valósul meg. Viszonylag kevéssé terjedt el adatátviteli célokra, mivel – bár gyakran előfordul, hogy a hasznos információ áramlása mindig azonos irányú – általában szükség van a vezérlőjelek ellenkező irányú továbbítására is. Egyszerű, kéthuzalos összeköttetéssel megvalósítható.

szimplex módszer (simplex method) – →lineáris programozási feladatok megoldási módszere: egyenértékű normál kanonikus rendszerek sorozatán olyan szélsőértékpontok bejárásával keresi az optimális megoldást, amelyekben a célfüggvény monotonan nő. A módszer gazdaságossága, hogy nem kell minden szélsőértékpontot megvizsgálni, csak azokat, amelyek egyre nagyobb célfüggvényértékeket adnak.

szimplex összeköttetés (simplex communication) – →adathálózat két →adatállomása közötti olyan stabil kapcsolat, amelyben az adatállomások adó-vevő viszonyban vannak egymással, az egyik mindig csak leadhat, a másik csak vehet, tehát a hasznos információ egy időben csak egy irányban haladhat a hálózatágon. A ~ csak a tényleges hasznos adatforgalomra vonatkozik, ugyanis a hasznos adatokon kívül az összeköttetés során vezérlőjelek is áramlanak a két állomás között. Ezek áramlása történhet az adatforgalommal egy időben v. vele váltakozva. Sokszor a vezérlőjelek külön felügyeleti csatornán áramlanak.

szimuláció (simulation) – 1. Valamely absztrakt v. valós jelenség dinamikus viselkedésére vonatkoztatott olyan matematikai →modell szerkesztése, amelynek változói a modellezett jelenséget mennyiségileg és minőségileg jellemzik. A matematikai modell paramétereinek számításával a szimulált jelenségre adhatnak számszerű összefüggést.

A számítógépes ~ főleg a sztochasztikus jelenségek ~jának megvalósításában hozott minőségi előrelépést, ugyanis ha a ~t a valóságnak megfelelő paraméterekkel elég sokszor hajtják végre, akkor az eredményekből a nagy számok törvénye alapján a valódi rendszerre vonatkozóan általában elegendő pontossággal kapható válasz a vizsgált kérdésekre. A hatvanas évek elejétől kezdve sorra jelentek meg a ~s programozási nyelvek (→SIMSCRIPT, →SIMULA, GPSS stb.). A számítógépes ~ megvalósításában általában öt jól elkülöníthető fázist lehet megkülönböztetni: a ~s feladat meghatározása és leírása; a matematikai modell előállítása; a modell számítógépes megvalósítása; a rendszer kipróbálása és az esetleges korrekciók elvégzése; a ~ gyakorlati megvalósítása, vagyis a modellezett folyamatra vonatkozó adatok, eredmények előállítása és kiértékelése. – 2. Valamely számítógép utasításainak másik számítógéprendszeren történő megvalósítása →mikroprogramok segítségével. Rendszerint meglévő számítógépen tesztelés céljából szimulálnak egy olyan számítógépet, amely még nem létezik, v. még csak a fejlesztés fázisában van, de tulajdonságai, jellemző paraméterei már meghatározottak. ~val egy-egy újabb számítógéprendszer →hardverje és →szoftverje párhuzamosan valósítható meg. – 3. Valamely más eszköz vezérlési sajátosságainak (pl. repülőgép) leutánzása program segítségével.

szimulációs aktualizálás (simulation-mode update) – →adatbázisoknál használha-

tó módszer, amelynél a tényleges →fizikai adatbázis adatelemeit nem változtatják meg, hanem az aktualizálni kívánt elemeket kimásolják, és azokon hajtják végre a tényleges aktualizálást. Az →adatintegritás biztosításában játszik fontos szerepet.

szimulációs nyelv (simulation language) – szimulációs eljárások programozására kifejlesztett magas szintű →programozási nyelv. Neumann János nevéhez fűződik annak a lehetőségnek a felismerése, hogy a számítógépek – igen nagy számú ismételt kísérlet lejátszásával – igen alkalmasak statisztikai információk gyűjtésére valamely jelenségre vonatkozóan. Az ismételt lejátszásokhoz szükséges véletlenszerűséget a programokba beépített →véletlenszám-generátor biztosítja.

Az első ~ek 1962-től kezdtek megjelenni (→SIMSCRIPT, →SIMULA, GPSS stb.). Specifikus utasításokkal rendelkeznek olyan jellemző műveletek leírására, mint a véletlenszám-generálás, a szimulációs idő vezérlése, a várakozási stratégiák programozása stb.

A ~ek két nagy csoportba sorolhatók. Az elsőbe tartoznak az eljárásorientált ~ek, amelyek a →modellek strukturális leírására alkalmasak, kiemelve a modellt jellemző erőforrásokat, amelyek az időben megjelenő felhasználók rendelkezésére állnak. A második csoportba tartoznak az eseményorientált ~ek, amelyeknél az események (felhasználók) időben jelennek meg, és várakozási stratégiák alapján jutnak az időben felszabaduló erőforrásokhoz.

szimulációs program (simulation program) – olyan program, amely bizonyos jelenség, folyamat, rendszer stb. időben bekövetkező állapotváltozásait írja le. Tkp. a szimulált objektum absztrakt modellje, amelynek alapján méréseket lehet végezni, megfigyeléseket lehet gyűjteni az illető objektumról. A felhasznált programozási nyelv rendszerint valamilyen specializált →szimulációs nyelv, de sok esetben használható az általánosan elterjedt magas szintű programozási nyelvek egyike is (pl. →FORTRAN, →PL/1 stb.).

Speciális felhasználásoknál a ~ olyan program is lehet, amelynek segítségével az adott számítógépen egy, még fejlesztés alatt álló számítógép működését szimulálják (→keresztszoftver).

szimultán működés (simultaneous operation) – 1. A számítógépek nagy részére →hardver szempontból jellemző működési mód, amelynél két művelet egyszerre, egymástól függetlenül végrehajtható, ha nem azonos erőforrásokat vesznek igénybe. Jellegzetes esete az olyan →periferikus művek működése, amelyek saját vezérlőművel rendelkező →adatcsatornákon csatlakoznak a számítógéphez. A →központi vezérlőegység csak előkészíti az adatátviteli művelethez szükséges információkat, de a tényleges adatátvitelt a vezérelt adatcsatorna önállóan végzi el, s így a központi vezérlőegység közben más műveleteket végezhet. – 2. Hibás értelmezésben a →multiprogramozással kapcsolatban is használják, ha több program látszólag egyidejűleg hajtható végre. Ez a ~ azonban csak látszólagos, ugyanis a központi vezérlőegység adott pillanatban tkp. csak egy program utasításait végzi, az →operatív memóriában található összes többi program felfüggesztett állapotban várja a központi vezérlőegység felszabadulását és rendelkezésre állását. Tényleges ~ csakis a →multiprocesszoros rendszerekben valósítható meg.

szinkron (synchronous) – olyan jelenség alapvető tulajdonsága, amelynek időben lefolyó állapotváltozásai egyértelműen függnek a viszonyítás alapjául szolgáló jelenség állapotváltozásaitól.

szinkron adatátvitel (synchronous communication) – távolsági adatátvitelben használt kommunikációs módszer, amelynél a néhány karaktertől több száz karakterig terjedő adatblokkokat folyamatos jelsorozat formájában viszik át. Az átvitel idején az adó- és a vevőállomás egymással →szinkronban működik. A tényleges adatátvitel jóval gyorsabb, mint az →aszinkron adatátvitelnél, mert nincs szükség a karakterenkénti start-, ill. stopjelekre. Bár az adatállomások berendezései bonyolultabbak és költségesebbek, mint az aszinkron adatátvitelnél, a ~ igen gazdaságos, elsősorban a közép- és nagy sebességű átviteleknél (így pl. a →számítógép-hálózatokban).

szinkron csatorna (synchronous channel) – olyan →kommunikációs csatorna, amely adatok és időzítő információk átvitelére

egyaránt alkalmas (nevezik még izokron csatornának [isochronous channelnek] is).

szinkronhálózat (synchronous network) – →adatátviteli hálózat, amelynél az összes kommunikációs kapcsolatot egy órajelhez szinkronizálták.

szinkronizálás (synchronization) – 1. →Konkurens folyamatok együttműködését szabályozó vezérlési módszer. A folyamatok az együttműködés során bizonyos erőforrások állapotának megváltoztatásával egymásnak információt adnak át. Ilyenkor lép fel az a követelmény, hogy egy folyamatnak (programnak) mindaddig felfüggesztett állapotban kell maradnia, amíg egy másik folyamat elő nem állította számára a működéséhez szükséges információkat. – 2. →Szinkron adatátvitelnél az →adatállomások előkészítése, ill. hangolása az átvitel megvalósítására. A leggyakoribb megoldás az, hogy az adatátvitelt néhány karakterből álló szinkronjel előzi meg, amely a vevőállomást az adóállomáshoz szinkronizálja. Egy másik módszer az, hogy meghatározott időközökben szinkronjelet illesztenek az adatok közé. Mindkét módszernél a szinkronjelsorozat máskor elő nem fordulható jelkombináció, amelyet a vevőállomás felismer.

szinkronizálási információ (synchronization information) – →adatátvitelben használatos (bináris karakterek formájában) kódolt információ, amelynek segítségével az adó-, ill. vevőoldali állomásokat szinkronizálják (→szinkron adatátvitel, →szinkronizálás, →kölcsönös szinkronizálás).

szinkronizálási karakter (synchronization character) – →adatátviteli hálózatokban használt speciális karakter, amelyet az adó- és vevőállomás →szinkronizálása céljából adnak a →kommunikációs csatornára.

szinkron (működésű) számítógép (synchronous computer) – olyan elektronikus számítógép, amelynél minden részművelet indítását adott órajel kezdeményezi, függetlenül az előző részművelet befejezésének időpontjától. A szinkron jelző nem az egyforma folyamatok egyidejűségét jelenti, hanem azt, hogy a különböző, egyidejűleg lefolyó műveletek azonos ritmushoz igazodnak. Noha a különböző folyamatok egyeztetése ilyenformán egyszerűbb, a korszerű számítógépek túlnyomó többsége →szinkron számítógép.

szintaktikai analízis (syntactical analysis) – programok →fordításának alapvető fázisa, amelynek elsőrendű célja a →forrásnyelvű programok nyelvi szerkezeteinek felismerése, valamint annak eldöntése, hogy ezek a szerkezetek elemei-e az adott →szintaxis által megadott nyelvnek. Járulékos feladatai közé tartozik az esetleges →szintaktikai hibák felismerése és kijelzése, a →szimbólumtáblázat felépítése, a fordítás következő fázisai számára megfelelő belső kód előállítása stb.

Tekintettel arra, hogy a programozási nyelvek szintaxisára ma már pontos leírás adható (pl. →Backus-Naur Forma), a ~ általában automatizált eszközökkel, viszonylag könnyen elvégezhető. A →generatív nyelvtan →helyettesítési szabályainak alkalmazása, ill. a →szintaxisfa felépítése alapján nemcsak az dönthető el egy mondatról, hogy hozzátartozik-e egy nyelvhez, hanem egyúttal kirajzolódik a mondat szerkezeti felépítése is, ami egyben a →kódgenerálás fázisának kiinduló információja.

A felhasznált stratégia alapján a ~ két osztálya különíthető el: a →felülről lefelé haladó (levezető típusú) szintaktikai analízis a →kezdőszimbólumból indul ki, és úgy próbál eljutni a vizsgált mondathoz, vagyis a szintaxisfát a gyökérelemből kiindulva építi fel; az →alulról felfelé haladó (összevonó típusú) szintaktikai analízis a szimbólumsorozatból indul ki, és megpróbálja azt visszavezetni a kezdőszimbólumra.

szintaktikai elemző (syntactical analyser) – a →fordítóprogram, ill. az →értelmezőprogram felépítésében részt vevő programegység, amelynek elsőrendű feladata a →szintaktikai analízis elvégzése, de járulékos feladatként a →forrásprogramot átalakítja a →kódgenerálás számára megfelelő belső kódra, felépíti a →szimbólumtáblázatot, kezeli és kiadja az esetleges →szintaktikai hibákat stb. Gyakorlatilag lehet önálló programegység, amely az egész forrásprogramra vonatkozólag elvégzi a szintaktikai analízist, és →korutinokat is alkothat a →lexikális elemzővel, ha minden újabb nyelvi elem szolgáltatásáért a lexikális elemzőhöz fordul.

szintaktikai hiba (syntactic error) – programozásban elkövethető hibatípus, amely a

választott →programozási nyelv standard előírásainak, szintaktikai szabályainak megszegéséből következik, s a nyelv felületes ismeretének következménye. A →fordítóprogramok egyik alapvető feladata, hogy a ~kat kiszűrjék és hibaüzenetek formájában a programozó tudomására hozzák.

szintaktikai kategória (syntax category) – valamely (programozási) nyelvhez tartozó olyan mondatrészek alkotják, amelyeknek ugyanaz a szerepük a →mondatok felépítésében. Így pl. a →Chomsky-hierarchia →generatív nyelvtanában mindazok a →terminális szimbólumláncok, amelyek ugyanabból a →nemterminális szimbólumból képezhetők, ~t alkotnak, s az illető nemterminális szimbólum nevét viselik.

szintaktikai kétértelműség (syntactical ambiguity) – olyan nyelvre vonatkozó tulajdonság, amelyben egy szintaktikai leképzés legalább két →mondatformát eredményez. →Környezetfüggetlen nyelvtanoknál ez azt jelenti, hogy az illető nyelvtan által generált nyelvhez tartozik, legalább egy olyan →mondat, amely legalább két →szintaxisfa alapján képezhető.

szintaktikai kiterjesztés (syntactic extension) – valamely →programozási nyelv gazdagításának módja, lényegében a nyelv →szintaxisának bővítése. Túl nagy gyakorlati jelentősége egyelőre nincs, esetleg a már létező programozási nyelvek kényelmesebb felhasználását teszi lehetővé. ~ pl. a →Pascal nyelvet megvalósító olyan →fordítóprogram, amelynek →kulcsszavai nem angol, hanem francia szavak.

szintaxis (syntax) – azoknak a szabályoknak az összessége, amelyeknek alapján adott nyelvhez tartozó helyes →mondatok előállíthatók, ill. a mondatok helyessége v. helytelensége kimutatható. A →fordítóprogramok az adott nyelvhez tartozó sorozatok felismerésére használják, és segítségével fedik fel az esetleges →szintaktikai hibákat.

Egy →programozási nyelv definíciója csak akkor teljes, ha ~át és →szemantikáját egyaránt megadják. A ~ leírására tudományosan megalapozott módszerek vannak, pl. a →formális nyelvek, a →Backus-Naur Forma és ezek továbbfejlesztett változatai.

szintaxisdiagram (syntax diagram) – valamely →programozási nyelv →szintaxisának leírására használható grafikus formalizmus.

szintaxisfa (syntax tree) – →környezetfüggetlen nyelvtanokban egy leképzési sorozatnak fastruktúrával ábrázolt formája. Gyökérelemét a →kezdőszimbólum (mondatszimbólum) alkotja, nemterminális csomópontjai a nyelvhez tartozó →nemterminális szimbólumok, míg a terminális elemek (levelek) a →mondatot felépítő →terminális szimbólumok. Minden nemterminális csomópontból irányított él mutat a →helyettesítési szabályban felhasznált alternatíva minden egyes eleméhez.

szintaxisvezérelt fordítóprogram (syntax-directed compiler) – a →fordítóprogramok előállításának automatizálása céljából kikísérletezett fordítóprogram-típus. Lényegileg olyan fordítóprogram, amely két fő részre osztható. Az egyik rész standard eljárások és táblázatok gyűjteménye, ez alkotja a fordítóprogram állandó, változatlan részét. A másik, a változó rész, az illető programozási nyelv →szintaxisának és →szemantikájának, ill. a →tárgynyelv specifikációjának formális, standard leírása. Bár a ~ok teljesítményszintje nem éri el az egyedi fordítóprogramokét, elsősorban a →hordozhatóság szempontjából előnyösek. Az egyik fontos, gyakorlatilag is megvalósított ~ a YACC.

szintetizált attribútum (synthetized attribute) – a D. Knuth által javasolt →attribútumnyelvtanban használt fogalom. Adott nyelvtan →nemterminális szimbólumához kapcsolt olyan attribútum, amelyet az illető nyelvtani egység a →terminális szimbólumoktól kijelölt attribútumokból és az →öröklött attribútumokból határoz meg, valamilyen leképző algoritmus segítségével.

szó (word) – 1. Önálló egységként értelmezhető, az adott számítógépre jellemző számú bitpozíciót tartalmazó vektor. A számítógép utasításai egy-egy ~ból v. a ~ egész számú többszöröséből állnak. →Bájtszervezésű számítógépeknél egy ~ rendszerint a →bájt páros számú többszöröse (legtöbbször 2, 4 v. 8 bájt). – 2. Valamely →programozási nyelv →szintaktikai kategóriája (→azonosító, →lefoglalt szó, →kulcsszó, →konstans stb.). Szövegszerűen elhatároló jelekkel (rendszerint szóközökkel v. írásjelekkel) köz-

revett karaktersorozat. Felismerése és kiemelése a →forrásnyelvű szövegből a →lexikális analízis alapvető feladata.

szoftver (software) – valamely számítógéprendszerhez tartozó programok, programrendszerek összefoglaló elnevezése, a →hardver mellett a rendszer másik alapvető összetevője. Első megközelítésben a ~ két részre osztható: a rendszerszoftver (alapszoftver) magában foglalja az →operációs rendszereket, a →fordítóprogramokat, a →kapcsolóprogramokat, a →betöltőprogramokat, a →kezelőprogramokat stb., általában mindazokat a programokat, amelyek a rendszer felhasználásában, ill. a felhasználói programfejlesztésben segédeszközként állnak a felhasználók rendelkezésére; az alkalmazói ~hez tartoznak a gépen megoldott gyakorlati feladatok programjai, programrendszerei.

szoftver erőforrás (software resource) – →programozott erőforrás

szoftverház (software house) – programtermékek előállításával, implementálásával, rendszertervezéssel és -fejlesztéssel foglalkozó vállalat.

szóhosszúság (word length/size) – adott számítógép egyik alapvető →hardverjellemzője, amely a gépi →szóban foglalt →bitek v. →bájtok számára vonatkozik. Fix ~nál a berendezés egész szerkezete – a szükséges →regiszterekkel együtt – és az →utasításkészlet összhangban van a rögzített hosszúsággal. Változó ~nál speciális mezőben kell megadni az aktuális utasítás hosszát.

szószerkezet (word structure) – adott számítógép egyik alapvető →hardverjellemzője, amely a konkrét gépi utasításszó szerkezetére (pl. hány bitpozíciót foglal el az utasításkód, a →címrész stb.), ill. az alapvető →[adat]típusok ábrázolási módjára vonatkozik (pl. →lebegőpontos számábrázolásnál hány bitpozíciót foglal el a →karakterisztika, hányat a →mantissza stb.).

szöveges mód (text mode) – →képernyők v. →nyomtatók egyik alapvető üzemmódja, amelynél a megjelenítési egység egy →karakter. A ~ segítségével legfeljebb igen egyszerű vonalas ábrák jeleníthetők meg (a másik üzemmód a →grafikus mód).

szövegkezdés karakter (start-of-text character) – →szinkron adatátviteli eljárásokban használatos vezérlőjel. A lekérdezett →terminál ~rel kezdi tényleges →üzenetét, és →szövegvége karakterrel fejezi be azt. →Üzenetkapcsolásnál a ~ helyén →fejléckezdet karakter áll, és csak utána következik a ~.

szövegszerkesztő (text editor, word processor) – szövegszerű információk előállítására, szerkesztésére szolgáló, általában →párbeszédes üzemmódban dolgozó segédprogram. Speciális alkalmazása a programszerkesztő, amely →forrásnyelvű programok szövegszerű összeállítását, →fordításra való előkészítését valósítja meg.

szövegvége karakter (end-of-text character) – →szinkron adatátviteli eljárásokban használatos vezérlőjel. A lekérdezett →terminál →szövegkezdés karakterrel kezdi és ~rel végzi →üzenetét.

sztochasztikus modell (stochastic model) – olyan matematikai →modell, amely a jelenségek véletlenszerű paramétereit a valószínűség-számítás, ill. a matematikai statisztika eszközeinek, módszereinek felhasználásával írja le.

szubrutin (subroutine) – utasítások olyan sorozata, amely adott program keretein belül önálló logikai és szerkezeti egységet alkot – rendszerint valamilyen részfeladatot végez –, s többször is végrehajtható a program különböző pontjain. Lehet nyílt, amikor az utasítássorozat mindannyiszor megjelenik a programban, valahányszor szükség van rá, és lehet zárt, amikor egy példányban szerepel, többszöri végrehajtása a program különböző pontjain hívások útján valósul meg. A legtöbb magas szintű →programozási nyelv implementációja lehetővé teszi ~oknak a főprogramtól különválasztott →fordítását, így azok →tárgyprogramja különböző főprogramokkal is kapcsolható (→kapcsolóprogram segítségével). Programozástechnikai szempontból a →moduláris programozás, ill. a →strukturált programozás alapvető eszköze.

szubrutinkönyvtár (subroutine library) – már kipróbált, bináris (esetleg közvetlenül végrehajtó) formájú →szubrutinok valamilyen →háttértárban tárolt rendezett gyűjteménye. Általában az →operációs rendszer standard eljárásait tartalmazza, amelyek a magas szintű →programozási nyelvek →fordítóprogramjai, ill. a →kapcsolóprogram közvetítésével

épülnek be a programok közvetlenül végrehajtható formájába. Ilyen eljárások pl. a különböző konverziós szubrutinok, a logikai adatátviteli műveleteket fizikai szinten megvalósító programegységek, a →standard függvényeket megvalósító szubrutinok stb.

szupervizor (supervisor) – →felügyelőprogram

szűrő (filter) – programozástechnikai értelmezésben olyan program, amely a bemenő adathalmazon viszonylag egyszerű átalakítást végez. Egymagában nem is volna jelentősége, viszont több hasonló típusú program sorozatba kapcsolása (vagyis adott – kimenő adathalmaza egyben a soron következő – bemeneti adathalmazát képezi) lehetővé teszi a →futószalagszerű feldolgozás elvének programozástechnikai megvalósítását.

szűrőállomány (query file) – →adatbázis-kezelő rendszerekben használt olyan speciális →[adat]állomány, amely az →adatbázis különböző kritériumok szerinti lekérdezését megvalósító szűrőfeltételeket tartalmazza. Használata lehetővé teszi, hogy adott [adat]állományból csak a konkrét szűrőfeltételnek megfelelő →rekordok vegyenek részt a tényleges feldolgozásban.

T

táblázat (table) – heterogén →lineáris adatstruktúra, amelynek elemeit legtöbbször nem a címük, hanem a tartalmuk alapján azonosítják. Egy-egy elemét több mező építi fel, ezek közül legalább egynek kitüntetett szerepe van, amelyet argumentumnak v. kulcsnak szoktak nevezni. A ~ elemeinek kikeresése a szerkezetből valamilyen →keresési algoritmus alapján végezhető el. Ha a ~ elemei rendezetlen formában vannak rögzítve, csak szekvenciális keresésre van mód, ha viszont argumentumok szerint rendezve vannak, több keresési algoritmus között is lehet választani (→lineáris keresés, →bináris keresés stb.).

táblázatban kereső utasítás (table look-up instruction) – egyes magas szintű →programozási nyelvekben olyan utasítás, amely a →keresés műveletét a →táblázat (→tömb) szintjén egyetlen szintaktikai szerkezet útján valósítja meg (pl. SEARCH a →COBOL-ban). Segítségével adott feltétel(eke)t kielégítő táblázatelem megkereshető a táblázatban (v. kimutatható a hiánya), és a megfelelő →index olyan értékre állítható, hogy a keresett elem közvetlenül elérhetővé váljon.

táblázatgenerátor (report generator/writer) – →COBOL programozási nyelvbe beépített automatizmus az adatfeldolgozási eredménytáblázatok előállítására. A felhasználó speciális bejegyzésekkel elsősorban a táblázat formáját, az adatelemeknek a táblázaton belüli elrendezését, nyomtatási formáját határozza meg, és viszonylag keveset kell törődnie a táblázatelemek előállítását végző eljárásokkal.

táblázatkezelő (spreadsheet) – →személyi számítógépekre kifejlesztett speciális programtermék, amely numerikus adatok táblázatos feldolgozását támogatja. A táblázat kölcsönösen függő sorokba és oszlopokba elhelyezett cellákból áll, amelyek közvetlenül v. más cellák tartalmának függvényében kaphatnak értékeket. Legfontosabb alkalmazási területei a gazdasági-kereskedelmi számítások, előrejelzések, tervezések, ütemezések stb. A legrégibb ilyen típusú termék az Apple cég által kifejlesztett VISICALC, más ismertebb változatai a Lotus 1-2-3, a Multiplan, a Supercalc stb.

takartvonal-algoritmus (hidden-line algorithm) – háromdimenziós grafikának kétdimenziós ábrázolásánál (pl. képernyőn) használt algoritmus, amely meghatározza, hogy bizonyos látószögekből mely vonalaknak nem szabad látszaniuk.

táregység (volume) – →kötet

tárgyköri szakértő (domain expert) – olyan elismert szaktekintélye az adott témakörnek, aki tanulás és az évek során szerzett gyakorlati tapasztalatok alapján kiemelkedő feladatmegoldási képességre tett szert, és aki képes és hajlandó ismereteit, gyakorlati módszereit, fogásait az →ismerettechnológussal együttműködve megfogalmazni, rendszerezni és formalizálni. Alapvető jellemzője, hogy az igen bonyolult feladatok megoldásánál – tapasztalata révén – képes már eleve kizárni azokat a lehetőségeket, amelyek nagy valószínűséggel nem vezetnek majd eredményre, s így elfogadható időn belül eljut a tényleges megoldáshoz. Meghatározó szerepe van a →szakértőrendszerek megvalósításában.

tárgymodul (object module) – →forrásprogram v. forrásprogramegység →fordításából származó programrész: programmodul v. adatmodul. Kódolási formája rendszerint v. →abszolút bináris, v. →áthelyezhető bináris forma. A fordítóprogram általában megfelelő típusú →könyvtárba rögzíti, ahonnan közvetlenül elérhető a →kapcsolóprogram számára.

tárgynyelv (object/target language) – a →fordítási tevékenységgel kapcsolatos szakkifejezés. Az a programozási nyelv, amelyre a →fordítóprogram lefordít bizonyos programozási nyelven (forrásnyelven) megírt algoritmust (programot). Általában a konkrét számítógépre jellemző →gépi nyelv v. →asszembler nyelv, de bizonyos esetekben lehet magas szintű programozási nyelv is (→előfordítás).

tárgyprogram (object/target program) – az a →tárgynyelvű program, amelyet a teljes forrásnyelvű program fordítása útján nyernek. Alapvető jellemzője az, hogy megőrzi az eredeti program szemantikai jelentését (vagyis a fordítás a →jelentésmegmaradás elvének betartásával történik).

tárgyprogramszerkesztés (object program assembling) – a →fordítás műveletének egyik fontos része, ill. a →fordítóprogram

egyik alapvető fázisa. Elsőrendű feladata a lefordított →forrásprogramból nyert generált kód átalakítása →tárgynyelvű programmá. A ~ járulékos feladatait elsősorban az határozza meg, hogy milyen jellegű tárgyprogram előállítása a cél. Ha a tárgyprogram a →kapcsolóprogram számára készülő →áthelyezhető bináris formájú modul, akkor a ~nek elő kell állítania azokat a járulékos információkat is, amelyek alapján a kapcsolóprogram a →külső hivatkozásokat feloldja, a bináris modulok →áthelyezését megvalósítja. Ha a tárgyprogram közvetlenül végrehajtható formára készül, a fenti műveletek szintén a ~ hatáskörébe tartoznak. A legegyszerűbb az →asszembler nyelvű tárgyprogram előállítása, ennek viszont az a hátránya, hogy a további feldolgozásokhoz még az →asszemblert is munkába kell állítani.

tárolási sebesség (recording rate) – →adathordozók egyik alapvető jellemzője: az egységnyi idő alatt rögzíthető →karakterek, →bitek, ill. →képpontok számával fejezhető ki. →Mágnesszalagoknál és →mágneslemezeknél kb. 0.1-3 millió karakter/sec, a →hajlékonylemezeknél 100-200 Kbájt/sec, →kompaktlemezeknél pedig 20 000-100 000 karakter/sec értékek az általánosak.

tárolási sűrűség (recording density) – azoknak a jeleknek a száma, amelyek az →adathordozó hosszirányú, keresztirányú v. területi egységében elhelyezhetők. Elsősorban a →mágnesszalagnál van jelentősége, mert a ~nek határt szab az egyes jelek felismerhetősége, megkülönböztethetősége, az író-olvasó fej mérete stb. Az érvényben lévő szabványok szerint a mágnesszalagoknál 800 →bpi és 19 000 bpi közötti értékek használhatók. A →mágneslemezeknél csak átlagos ~ről lehet beszélni, ugyanis a belső →sávokon nagyobb, a külső sávokon kisebb ~ érhető el. Értéke 15 000 bpi-ig mehet el. →Hajlékonylemezeknél a ~ értékei 2768 bpi (SD), 5536 bpi (DD), ill. 9640 bpi (HD). Optikai adathordozóknál felületi ~ről beszélnek. Orientatív értékei: 450 karakter/mm^2 a mikrofilmeknél, kb. 30 000 bit/mm^2 a →kompaktlemezeknél stb.

tárolt program (stored program) – a →digitális számítógépek elvi működését ma is meghatározó, Neumann Jánostól származó elv, amelynek értelmében az adatok és az azokkal dolgozó utasítások egyaránt az →operatív memóriában találhatók, ugyanolyan kódolási formában és a →központi vezérlőegység szempontjából azonos elérhetőséggel. Ebből következik, hogy a program – önmagát adatként kezelve – saját állapotát is megváltoztathatja.

társprocesszor (coprocessor) – speciális kiépítettségű →mikroprocesszor, amely rendszerint egy →központi vezérlőegység szerepét játszó mikroprocesszorral együtt működik, annak funkcióválasztékát bővítve. A legismertebb típusú ~ az →aritmetikai társprocesszor, de alapjában véve a megvalósított funkciókészletnek csak technológiai korlátai vannak.

tartalék redundancia (stand-by redundancy) – a →megbízhatóság növelése céljából a számítógéprendszerbe több hasonló típusú funkcionális egység integrálása, amelyek közül adott pillanatban csak egy van ténylegesen aktív állapotban. Ha ez valamilyen hiba folytán kiesik, feladatát automatikusan átveszi egy tartalék egység (szemben a moduláris redundanciával).

taszk (task) – →feladat

táv-adatfeldolgozás (teleprocessing) – az →adatfeldolgozásnak az a módja, amikor a feldolgozó számítógép, ill. a →periferikus egységek földrajzilag különböző helyeken találhatók, s →kommunikációs csatornák biztosítják közöttük a közvetlen kapcsolatot. Két alapvető formája ismeretes. Azonos idejű ~nál a végberendezésekből a számítógépbe juttatott adatokat rögtön feldolgozzák, s a feldolgozás eredménye közvetlenül visszajut a végberendezésekhez. Kötegelt ~nál a végrehajtás időpontja független az adatok érkezésétől, s az eredmények kiadása is eltolódhat időben a feldolgozástól.

Szervezési és rendszertechnikai szempontból a ~ az adatfeldolgozással és az adatátvitellel kapcsolatos műveletek, módszerek rendszere.

távfeldolgozási mód (telecommunication access method) – a →soros távfeldolgozási módhoz hasonló, de annál több szolgáltatást nyújtó elérési mód a →táv-adatfeldolgozású rendszerekben. Lehetőséget ad közvetlen kapcsolatra is a →terminálok és a központi

feldolgozó számítógép között, s gyors átviteli módokat tud biztosítani. Használata azt is lehetővé teszi, hogy a terminál egy másik távfeldolgozó rendszer legyen. ~ban a →b/k műveletek nem csak egy várakozási soron bonyolódnak le, így az alkalmazási programok közvetlen író-olvasó →makroutasításokat is használhatnak.

távolsági kötegelt feldolgozás (remote batch processing) – a →táv-adatfeldolgozás olyan megvalósítása, amelynél a feldolgozó nagyszámítógép és a →munkák bevitelére szolgáló, ill. a feldolgozás eredményeit kiadó végberendezések földrajzilag távol vannak egymástól. A munkákat a kihelyezett standard →bemeneti egységen (pl. →kártyaolvasó) keresztül juttatják el a központi számítógéphez. Komplexebb rendszerekben a munkák már a végberendezéseknél bizonyos előfeldolgozáson esnek át (forrásprogram- szerkesztések, bemeneti adatok érvényesítése stb.). A végállomásokkal kapcsolatos adatforgalom lebonyolítására az →operációs rendszert adatátviteli modulokkal kell kiegészíteni. A ~ lényegében csak abban különbözik a klasszikus helyi →kötegelt feldolgozástól, hogy a munkák felvétele a végállomásokról történik, s az eredmények szétosztása, végállomásokra történő irányítása a munkavezérlő utasításoknak megfelelően valósul meg. Természetesen a →munkavezérlő nyelvet a végállomások kezelésére vonatkozó utasításokkal kell kiegészíteni.

A ~sal párhuzamosan a központi számítógépen rendszerint helyi kötegelt feldolgozás is végezhető, a munkák ütemezésénél már nem veszik figyelembe azok származási helyét.

távvezérelt periféria (teleprocessing peripheral equipment) – valamely számítógéprendszerhez közvetlenül (→on-line) kapcsolódó olyan →periféria, amely a központi számítógéptől földrajzilag távoli helyen található, s ahhoz →adatátviteli hálózaton keresztül kapcsolódik, de éppen úgy a számítógép vezérlése alatt áll, mint a helyi periféria.

teknőcgrafika (turtle graphics) – a →számítógépes grafika egyik, elsősorban oktatási célokat támogató módszere, a →LOGO →programozási nyelv készítői vezették be. Lényege az volt, hogy egy sík felületű rajzgép tollát hordozó kocsit (teknőcöt) megfelelő parancsok segítségével mozgatnak, s így különböző rajzolatok, grafikus ábrák készíthetők. A módszert továbbfejlesztették a →képernyős megjelenítők számára is: a tollat hordozó kocsit felváltotta egy képernyőn megjelenített, háromszög alakú nyíl, amely mozgatása révén nyomot hagy maga után a képernyőn.

teletex (teletex) – közepes és nagy sebességű (2400 bps) nyilvános →kapcsolt hálózat szöveges információk forgalmazására (CCITT szabvány), amely lassan teljesen felváltja a hagyományos →telexet.

telex (telex) – alacsony sebességű (50 bps) nyilvános →kapcsolt hálózat, amelyet világszerte adminisztrációs (szöveges) üzenetforgalmazásra használtak. Napjainkban egyre jobban kiszorítja a →teletex.

teljes duplex csatorna (full duplex channel) – a →duplex csatorna elnevezése a →félduplex csatornától való megkülönböztetés céljából.

tényleges cím (effective address) – az a →cím, amely – az esetleges →indirekt címzési lehetőségeket is figyelembe véve – az adat →operatív memóriabeli lokalizálását lehetővé teszi.

tényleges kulcs (actual key) – →közvetlen hozzáférésű [adat]állománynál az a változó, amely az [adat]állomány egy-egy →rekordjának visszakereséséhez, lokalizálásához szükséges információt tartalmazza. A felhasználó és a rendszer kapcsolata szempontjából úgy tekinthető, mint egy olyan adatnév, amelyhez mind a programozó, mind a közvetlen hozzáférésű →periferikus egységet vezérlő rendszerprogram hozzáfér. A ~ nem tartozik közvetlenül a →logikai rekordhoz, aktuális értékét a felhasználói program biztosítja. Tartalma a →terítőfüggvény kiinduló adata, amelyből a megfelelő leképzés útján előállítja a keresett rekord fizikai címét az adathordozón.

terítőfüggvény (randomizing function, hashing) – →közvetlen hozzáférésű [adat]állományoknál a közvetlen címzés megvalósítására használatos algoritmus (függvény), amely adott →tényleges kulcsból az illető →rekord fizikai címét leképezi. Ez a leképzés a gyakorlatban csak megközelítően egyértelmű, több tényleges kulcshoz rendelheti ugyanazt a fizikai címet (szinonimákat állít

elő). A konkrét alkalmazástól függ, mennyire felel meg egy-egy ~. A legismertebbek a következő eljárások egyikét használják:

1) maradékmódszer: a kulcs értékét elosztják adott M modulussal, és a maradék lesz a relatív →sáv- v. →blokkcím; a jó eloszlás érdekében célszerű modulusként prímszámot használni;

2) osztásmódszer: a kulcsot elosztják adott M számmal, s a hányadost használják relatív címként;

3) folding: a kulcsot több részre osztják, és ezeket összeadják; a műveletet esetleg többször is meg lehet ismételni;

4) csonkítás: a kulcsból bizonyos (rendszerint szélső) jegyeket elhagynak;

5) alaptranszformáció: a tízes számrendszerben felírt kulcsot a jegyek felhasználásával más alapú számrendszerben felírtnak tekintik, s így számítják ki az új értéket;

6) szorzás módszer: a kulcsot megszorozzák az adott számmal, majd esetleg további alakításokat végeznek;

7) osztás polinommal: a kulcs jegyeit egy polinom együtthatóinak tekintik, s ezt osztják egy rögzített polinommal, a maradék polinom jegyei lesznek a cím jegyei.

terminál (terminal) – az ember-gép határvonalon dolgozó, a →távvezérelt perifériához tartozó berendezés, amely saját vezérlőegységgel rendelkezik, s lehetővé teszi valamely, esetleg földrajzilag távoli számítógéprendszer felhasználását. A számítógép és a ~ közötti kapcsolat rendszerint →adatátviteli hálózaton keresztül valósul meg, de ha a távolság megengedi, kapcsolódhat közvetlenül is a számítógéprendszerhez. Jellegzetes felhasználási területe a →párbeszédes üzemmódban dolgozó számítógéprendszer.

~ként kisszámítógép is működhet (esetleg idejének csak egy részében), s az ilyen →intelligens terminál csak akkor fordul a központi számítógéphez, amikor feladatai megoldásához saját →erőforrásai nem kielégítőek.

terminális szimbólum (terminal symbol) – a →formális nyelvek elméletében azokat az alapszimbólum-sorozatokat nevezik így, amelyekből adott nyelvhez tartozó mondatstruktúrák felépíthetők (megjegyzendő, hogy egy nyelv nem tartalmazza az ábécéjéből alkotható összes lehetséges alapszimbólum-sorozatot). A legtöbb →programozási nyelv a ~ok következő típusait használja: →lefoglalt szó, →kulcsszó, →literál, →konstans, →operátor, betűk, számok, írásjelek stb.

A természetes nyelvek viszonylatában a ~ok halmaza az illető nyelvek szókincsének felelne meg (→generatív nyelvtan).

tesztelés (testing) – →programtesztelés

tesztelési adatbázis (testing data base) – valamely konkrét →adatbázis szerkezetének megfelelő, de csak kevés →entitást tartalmazó adatbázis, amely a felhasználói programok kipróbálására szolgál. Így nem az élő adatbázist veszélyezteti az esetleg még hibásan működő program. A legtöbb →adatbázis-kezelő rendszer rendelkezik olyan segédprogrammal, amely lehetővé teszi a ~ automatikus generálását.

tesztgenerátor (test generator) – egyes →operációs rendszerekben olyan segédprogram, amely a felhasználói programfejlesztésben jelent hatékony segédeszközt. Lényegében olyan program (rendszer), amely a felhasználói specifikációk alapján a konkrét felhasználásban feldolgozandó →[adat]állományok felépítéséhez és szerkezetéhez hasonló [adat]állományokat képes előállítani, s így a felhasználói programot alapos tesztelésnek lehet alávetni még az élesfutás megkezdése előtt. A ~ jelentősége az, hogy általa olyan határesetnek számító →logikai rekordok is generálódnak, amelyek élesfutásnál esetleg csak hosszabb periódus eltelte után jelentkeznek, s így csak későn derül fény a rejtett logikai következetlenségekre. Ez igen komoly következményekkel járhatna a konkrét felhasználói [adat]állományok állapotára nézve.

tesztprogram (test program) – 1. Speciális rendeltetésű program, amelynek segítségével egyes →hardverelemek helyes működését ellenőrzik v. esetleges hibák behatárolását és kimutatását végezhetik. – 2. →Fordítóprogramok, ill. →értelmezőprogramok tesztelésére, érvényesítésére készült speciális program, amely egy-egy szintaktikai szerkezet, ill. a kapcsolt szemantika helyes kezelésének ellenőrzésére szolgál.

tetrád (nibble) – négy egymás utáni →bitből álló egység, amely egy decimális szám ábrázolásához szükséges. Tekintettel arra, hogy egy ~ 16 különböző kódolási lehetősé-

get biztosít, a decimális számok viszont csak 10 különböző kombinációt igényelnek, 6 olyan ~ marad, amelyekhez nem rendelnek decimális jegyet. Ezeket →áltetrádoknak nevezik, s hibaellenőrzésre használhatják.

tevékenységjegyzék (action entries) – a →döntési táblázat jobb alsó része: az egyes feltételkombinációkhoz tartozó tevékenységek összessége. Szavakkal kifejezve a következő kérdésre ad választ: „Melyek az adott feltétel(ek)nél végrehajtandó tevékenységek?"

tevékenységtörzs (action stub) – a →döntési táblázat bal alsó része: a leírt rendszer olyan elemeinek összessége, amelyek végrehajtása a rendszer (algoritmus) viselkedésével együtt jár. Szavakkal kifejezve a következő kérdésre ad választ: „Melyek a szóba jöhető tevékenységek?"

típusegyezés (type compatibility) – →erősen típusos nyelvekre jellemző azon →szemantikai szabályok összessége, amelyek meghatározzák, hogyan használhatók, kombinálhatók a különböző →[adat]típusú elemek (→változók, →literálok, →függvények stb.) ugyanabban a kifejezésben, értékadásban v. →paraméterátadásban. (Érdekes módon a →Pascal, a →Modula, a →CHILL, ill. az →Ada nyelvek alapvető eltéréseket mutatnak a ~ előírásainak tekintetében.)

típusfeltétel (class condition) – →COBOL programozási nyelvben használatos →egyszerű feltétel, amely implicit v. explicit módon DISPLAY tárolási formában deklarált →változó aktuális értékére vonatkozik (NUMERIC v. ALPHABETIC).

típusteszt (class test) – →típusfeltétel

tiszta kód (pure code) – olyan formában megírt program, amelynek egyik része sem változik meg a végrehajtás alatt. Ez úgy valósul meg, hogy az összes adat- és munkatár a program területén kívülre esik, s a program nem változtatja meg önmagát. A ~ot használó folyamat bármely ponton megszakítható, és egy másik folyamattal a program újraindítható. Az →osztott erőforrásként használt →felügyelőprogram szubrutinjainak feltétlenül ilyen formájúaknak kell lenniük. A ~ nagymértékben javíthatja a →lapkezelés hatékonyságát is, ugyanis az ilyen lapokat felülírhatják anélkül, hogy előzőleg →háttértárba mentették volna (→újraindítható program).

torlódás (congestion) – →adatátviteli hálózatokban a továbbítható adatmennyiség korlátainak túllépésekor bekövetkező állapot. ~kor a →vonalkapcsolásos hálózatokban lehetetlenné válnak a további hívások, a →csomagkapcsolásos hálózatokban pedig megnő a továbbítási késleltetés.

többfázisú összeválogatás (polyphase merging) – az összeválogatásnak a külső →rendezésre alkalmazott módszere. Lényege, hogy a rendelkezésre álló →periferikus egységek közül egyet üresen hagynak, a többire pedig szétosztják a teljes rendezendő adathalmazt úgy, hogy egyes perifériákra különböző számú rendezett részhalmaz jusson. Ezeket összeválogatják az üres egységre egészen addig, amíg az egyikről el nem fogynak a rendezett részhalmazok. Azután erre folytatják az összeválogatást, amíg egy újabb periferikus egység ki nem ürül. Ettől kezdve erre történik az összeválogatás, egészen addig, amíg a teljes rendezett adathalmazhoz nem jutnak.

többletráfordítás (overhead) – az a többletidő, amelyet a számítógép →központi vezérlőegysége nem kimondottan a felhasználói program végrehajtására, hanem saját →erőforrásainak adminisztrálására használ. Megközelítően az az időkülönbség, amely adott program többfelhasználós környezetben történő végrehajtására szükséges, azzal az idővel szemben, amelyre akkor lenne szükség, ha az egész rendszer kimondottan az illető program kizárólagos rendelkezésére állna. ~t eredményez az erőforrások kiosztása, a különleges események (→megszakítások) feldolgozása, az állapotinformációk adminisztrálása, az adatvédelmi tevékenység, a számlázás stb.

többprocesszoros rendszer (multiprocessor system) – olyan számítógéprendszer, amelynél a nagyobb hatásfok és működési biztonság érdekében több →processzor működik közre a feladatok végrehajtásában. A →multiprocesszorral szemben a ~ben mindegyik processzor saját →operatív memóriával rendelkezik, bár szükséghelyzetben egymás memóriáit is elérhetik. A ~ működését egyetlen →operációs rendszer vezérli. A pillanatnyi igényeknek és az →erőforráskészletnek megfelelően az operációs rendszer osztja ki a fel-

adatokat a közreműködő processzoroknak, s ugyanazt a feladatot párhuzamosan végeztetheti két v. több processzorral, a nagyobb működési biztonság érdekében.

többszörös hozzáférés (multiaccess) – számítógéprendszerekben az a hozzáférési mód, amikor a rendszer →erőforrásait egymástól függetlenül egyszerre több felhasználó is igénybe veheti. Látszólagos a ~ pl. az →időosztásos rendszerekben, ahol az egyes felhasználók tkp. csak sorra és egyenként jutnak a rendszer erőforrásaihoz. A rendszer működési sebessége azonban annyira meghaladja a felhasználók reakcióképességét, hogy a számítógép látszólag egyszerre szolgálja ki az összes felhasználót. Ugyancsak látszólagos a ~ az →adatbázisoknál, ahol az →adatbázis-kezelő rendszer lehetővé teszi, hogy több felhasználó is hozzáférhessen az adatbázisban tárolt információkhoz. A tényleges ~t csak a →multiprocesszoros rendszerek, ill. a →többprocesszoros rendszerek képesek biztosítani.

töltőkarakter (pad character) – jelentés nélküli karakter, amelyet a →fizikai bejegyzések, →üzenetek, →adatblokkok stb. végére illesztenek, hogy az előírt fix hosszúságot biztosítsák.

tömb (array) – homogén (hasonló attribútumokkal rendelkező), egyenlő hosszúságú elemekből felépülő →lineáris adatstruktúra, amelynek elemei a struktúrában elfoglalt relatív helyük alapján érhetők el. Az →operatív memóriában szekvenciálisan ábrázolják: sorfolytonosan (a ~ sorait helyezik el hézagmentesen) v. oszlopfolytonosan (a ~ oszlopait helyezik szekvenciálisan, hézagmentesen).

A ~ lényegében csak abban különbözik az egydimenziós lineáris adatstruktúrától, hogy az adatelemek kiválasztása két (v. több) →index alapján történik. A szekvenciális ábrázolás itt is optimális a helyfoglalás szempontjából, mert a sorrendi összefüggéseket nem kell járulékos adatként tárolni (nincs szükség →láncolásra, ill. →mutatókra), az adatelemek relatív címe egyszerű aritmetikai összefüggés alapján meghatározható. A magas szintű →programozási nyelvek nagy többsége deklaratív és feldolgozó utasításokkal rendelkezik a ~ök előállítására és kezelésére. A →fordítóprogram feladatkörébe tartozik, hogy az →indexes változó címének kiszámítására megfelelő →gépi utasítássort generáljon. A ~ szerkezeti kialakítása az →asszembler nyelvekben is lehetséges, de a címszámítások a felhasználóra hárulnak.

tömb [adat]típus (array [data] type) – →erősen típusos nyelvekben definiált, rögzített számú homogén adat együttes kezelésére használható →strukturált [adat]típus. Értékkészlete megegyezik az összetevő elemek közös értékkészletével, megengedett műveletei az értékadás és az elemkijelölés. Az adatstruktúrát felépítő elemek maguk is lehetnek összetett struktúrák, a szelektor (→index) viszont csak →sorszámozott [adat]típusú lehet. A ~ egyszerűbb változata (a →tömb) egyes klasszikus nyelvekben (pl. →ALGOL, →FORTRAN, →PL/1 stb.) is megtalálható.

tömegtár (mass memory) – igen nagy kapacitású, de ugyanakkor elég nagy hozzáférési idejű, közvetlenül címezhető →memória, amelynek alapvető jelentősége az igen alacsony tárolási költség. Rendszerint az →operatív memória kiegészítéseként használják a különböző, gyakran igényelt →rendszerprogramok tárolására. Jellegzetes ~ pl. a →mágneslemez.

tömörített decimális ábrázolás (packed decimal representation) – decimális számok ábrázolására használt olyan forma, amelynél minden →bájt két-két decimális jegyet tartalmaz. A jobb szélső bájt tartalmazza a decimális szám utolsó jegyét és az előjel számára fenntartott →tetrádot: a + ábrázolására a hexadecimális A, C, E és F, a – számára a B és D használatosak. Elsősorban a decimális aritmetika műveleteit végzik a ~ú operandusokkal, a →b/k egységekben a →zónázott decimális ábrázolást használják. A decimális →aritmetikai egységgel rendelkező számítógépek →utasításkészletében mindig megtalálható az a speciális utasításpár, amely e két decimális ábrázolási formátum közötti átalakításokat elvégzi.

törlő olvasás (destructive reading) – egyes memóriatípusokra (elsősorban a →ferritmemóriára) jellemző tulajdonság, amelynek lényege az, hogy az olvasási művelet nyomán az adathordozó elveszti eredeti tartalmát.

törölhető-programozható permanens memória (erasable programmable read-only memory) – olyan programozható →perma-

nens memória, amelyben ultraibolya sugarakkal az előzőkben beírt információ törölhető (→off-line üzemmódban), s újra programozható állapotba hozható.

törzs[adat]állomány (master file) – valamely felhasználói program alapvető adatait tartalmazó →[adat]állomány, amely az illető felhasználás pillanatnyi állapotát tükrözi, →entitásai viszonylag állandó jelleggel rendelkeznek, de a hozzájuk rendelt tulajdonságok időben változhatnak. Jellegzetes ~ pl. egy vállalat összes alkalmazottjának adatait tartalmazó [adat]állomány.

transzparens adat (transparent data) – a fizikai adatszervezés megvalósítására használt olyan járulékos adat, amely csak a fizikai adatleírás szintjén létezik, a programozónak a logikai adatfeldolgozás szintjén nem is kell tudomást vennie róla. ~ok pl. a →lista jellegű adatstruktúrák megvalósítására használt →mutatók, az →indexszekvenciális [adat]állományok fizikai tárolását rögzítő indextáblázatok, az →állapotinformációk, az adatátvitelben megjelenő vezérlőkarakterek stb.

transzparens erőforrás (transparent resource) – az →operációs rendszer olyan programozott →erőforrása, amely lehetővé teszi az adatfeldolgozás logikai, ill. fizikai szemléletének elhatárolását. A ~ úgy aktivizálódik, hogy arra a programozónak nem kell explicit igényt kifejeznie. Jellegzetes ~ok pl. az →[adat]állományok logikai feldolgozását biztosító →[adat]állomány-kezelő rendszer szubrutinjai, amelyek a programozó logikai szintű adatleírásait, ill. utasításait átalakítják az adattárolás fizikai valóságára. Így →indexszekvenciális [adat]állománynál a programozónak elég az →aktuális kulcs, ill. a →logikai rekord leírását megadnia ahhoz, hogy munkaterületén a kért logikai rekord megjelenjék, s teljesen „átlátszó" lesz számára az az egész sor konkrét, eléggé bonyolult keresési és átviteli művelet, amelynek révén az illető rekord ténylegesen eljut az adathordozóról a meghatározott munkaterületre.

tranzakció (transaction) – számítástechnikai értelmezésben a valóságban történő valamely eseményt, változást rögzítő adatok összessége. →Adatbázisokkal kapcsolatban meghatározott adatkezelési művelet indítási igényének leírása.

tranzakciófeldolgozás (transaction-oriented processing) – →adatbázisokkal kapcsolatos olyan feldolgozási mód, amikor az egyes →tranzakciókat jelentkezésük után közvetlenül feldolgozzák. Gyors válaszidőt igénylő, eseményorientált felhasználások jellegzetes feldolgozási módszere, amely különös jelentőségre tett szert az →elosztott adatbázisok, a →számítógép-hálózatok, általában az →elosztott feldolgozás megjelenésével.

tudományos programkönyvtár (scientific library) – a legtöbb általános célú számítógéprendszerben megtalálható →könyvtár, amely matematikai algoritmusok, módszerek programozott gyűjteményét tartalmazza, általában →paraméterezett eljárások formájában.

túlcsordulás (overflow) – →aritmetikai utasítások végrehajtása nyomán előadódható állapot, ha a műveletek eredménye nem ábrázolható helyesen (nagyobb, mint a →regiszterben v. a memóriarekeszben ábrázolható legnagyobb érték). A konkrét számábrázolásnak megfelelően lehet fixpontos (bináris), lebegőpontos v. decimális. Megjelenése általában →csapdához vezet, de a felhasználó →maszkolással semlegesítheti a program megszakítását (természetesen ilyenkor magának kell felülvizsgálnia számítási eredményeit).

túlcsordulásbit (overflow bit) – az →aritmetikai egység egyetlen bitpozíciót tartalmazó →regisztere, amely az aritmetikai utasítások végrehajtása nyomán a legnagyobb helyértéken fellépő átvitelt (→túlcsordulást) tartalmazza. A számítógépek utasításkészletében mindig szerepel a ~ tartalmától függő feltételes utasítás. Megjegyzendő, hogy a ~ 1-es értéke nem jelent mindig feltétlenül túlcsordulást. Pl. →komplemens kódban két negatív szám összege minden esetben 1-re állítja a ~et, mégsem jelent túlcsordulást.

U, Ú

ugró utasítás (branch instruction/statement) – olyan →gépi nyelvű v. magas szintű utasítás, amely a program szekvenciális végrehajtását megszakítja s a vezérlést a program egy másik részében található utasításra adja. Az – lehet feltételhez kötött, amikor a vezérlésátadást csak bizonyos feltétel megvalósulása válthatja ki, ellenkező esetben a végrehajtás szekvenciálisan folytatódik, de lehet feltételfüggetlen is, amikor a vezérlésátadás mindenképpen megvalósul.

újrahasználható adathordozó (reusable media) – olyan (rendszerint mágneses) →adathordozó, amelyen a tárolt információk törlése után újabb adatokat rögzíthetnek. A legfontosabb –k közé tartozik a →mágneslemez, a →mágnesszalag, a →mágneskazetta, a →hajlékonylemez stb.

újraindítás (restart) – 1. Valamely hiba miatt megszakadt feldolgozási folyamat folytatása az utolsó →újraindítási pontból. – 2. →Újraindítható programok többszöri elvégzése más és más adathalmazokra vonatkoztatva.

újraindítási pont (restart point) – feldolgozási folyamatok során az →operációs rendszer által automatikusan v. a felhasználói program kezdeményezésére készített állapotinformációk összessége, amelyek segítségével a bekövetkezett hiba elhárítása után a korábbi helyes állapot visszaállítható és a folyamat újraindítható. A periodikusan megmentett állapotinformációk számára a legtöbb operációs rendszerben speciális →rendszerállományok állnak rendelkezésre, de felhasználói →[adat]állományok is használhatók erre a célra. Az –ok használata az →adatbiztonság, ill. az →adatintegritás megvalósításának alapvető módszere.

újraindítható program (reentrant program) – olyan program, amely megfelel a következő alapvető követelményeknek: nem módosítja önmagát és nem helyez el adatokat a saját területén (tehát →tiszta kódolású program). Általában a több felhasználó által is igénybe vett programok, pl. a →felügyelőprogram szubrutinjai kell hogy rendelkezzenek ezzel a tulajdonsággal, mert csak így elegendő, hogy egyetlen példányuk legyen jelen az →operatív memóriában.

Az – a →lapkezelés hatékonyságát is javíthatja, ugyanis az ilyen programokat megszakítás után és felülírás előtt nem szükséges →háttértárba kimenteni, mert az eredeti példány mindig rendelkezésre áll az újbóli betöltésre.

újraindíthatóság (reentrancy) – valamely programegységnek az a tulajdonsága, hogy adott →címtartományra vonatkoztatott végrehajtása felfüggeszthető, egy másik címtartományra vonatkoztatva újra aktivizálható, majd végrehajtása később a felfüggesztési pontból az eredeti címtartományon folytatható. Az illető programegységnek tehát újraindítható programnak kell lennie. A →felügyelőprogramhoz tartozó programegységeknek rendszerint rendelkezniük kell ezzel a tulajdonsággal, hogy egyetlen példányuk kiszolgálhasson több felhasználót is. Ezek általában →tiszta kódolású programok, amelyek adataikat mindig a felhasználó adatterületén helyezik el.

újraprogramozható permanens memória (reprogrammable read-only memory) – →törölhető-programozható permanens memória

újraszerkesztés (restructuring) – az →adatbázis logikai szerkezetének módosítása annak következtében, hogy a valós világ szemantikai összefüggései v. a feldolgozási igények megváltoztak.

újraszervezés (reorganisation) – 1. Az →adatbázis fizikai szervezésének módosítása – általában hatékonyságnövelő célokból – a logikai szerkezet változtatása nélkül. Az →adatfüggetlenség által biztosított legfontosabb lehetőség. – 2. Az →[adat]állományok karbantartásának fontos művelete, elsősorban az →indexszekvenciális [adat]állományok v. a →direkt szervezésű [adat]állományok használatánál. Célja az, hogy a leggyakrabban használt adatokat az elsődleges adatterületre tömörítsék, s így a hozzáférés hatékonyságát javítsák. Végrehajtásához az →operációs rendszerek általában specializált →kezelőprogramot állítanak a felhasználók rendelkezésére.

újsor karakter (line feed character) – képernyős →megjelenítő, ill. →nyomtató számára fontos →vezérlőkarakter. Hatására a következő megjelenítendő karakter egy sorral lejjebb, a sor bal első pozíciójába kerül. Nyomtatónál

a parancs hatására a papír fizikailag is elmozdul a sorokra merőlegesen, képernyőn egysornyi függőleges →görgetéstörténik,amenynyiben a →helyőr az utolsó sorban áll.

univerzális kvantor (universal quantifier) – →predikátum

UNIX – a Bell Laboratóriumban a hetvenes évek elején kifejlesztett →operációs rendszer, amely teljes egészében a →C programozási nyelvre épül. Először a DEC PDP-11 típusú miniszámítógépeken üzemelt, majd nemsokára – igen előnyös jellemzőinek köszönhetően – elterjedt szinte az összes 16, ill. 32 bites számítógépen. Többfelhasználós →multiprogramozást támogató önálló v. alárendelt operációs rendszerként is működhet, →parancsértelmezője (SHELL) igen barátságos →párbeszédes üzemmódú környezetet biztosít felhasználói számára. Legismertebb változatai a Siemens által kifejlesztett SINIX, ill. az IBM →személyi számítógépeire, a Microsoft cég által létrehozott XENIX.

UPC (Universal Product Code) – egységes termékkód, az európai →EAN-kód amerikai megfelelőjeként használt →vonalkód.

utasítás (instruction, statement) – statikus szemléletben olyan szimbolikus jelsorozat, amelynek segítségével egy számítógép által elvégzendő műveletet – valamely →programozási nyelv formalizmusával – leírnak. Dinamikus szempontból a számítógép által elvégzett tényleges művelet (elemi műveletsor). Szintaktikai szempontból a programozási nyelvek legkisebb funkcionális egysége, →mondata, amelynek segítségével algoritmusokat, eljárásokat lehet kódolni programok alakjában. Az ~ az →asszembler nyelvek szintjén egyetlen →gépi utasításnak felel meg, a magas szintű programozási nyelvek szintjén viszont rövidebb-hosszabb gépi utasítássorozatot jelent.

utasításaktivizálás (instruction activation) – a →központi vezérlőegység által végrehajtott elemi műveletek sorozata, amelyek előkészítik egy →gépi utasítás tényleges végrehajtását. Két alapvető fázisra épül: az →utasításlehívásra és az →utasításértelmezésre.

utasításciklus (instruction cycle) – 1. Azoknak a műveleteknek az összessége, amelyeket az →utasításaktivizálás és a tulajdonképpeni végrehajtás céljából a →központi vezérlőegység elvégez. – 2. Az az időintervallum, amely egy →gépi utasítás aktivizálásához és tulajdonképpeni végrehajtásához szükséges.

utasításdekódoló (instruction decoder) – a →központi vezérlőegységnek az a része, amely a →gépi utasítás műveleti kódrészét értelmezi és ennek alapján az illető utasítást ténylegesen megvalósító további elemi műveleteket kiváltja.

utasításértelmezés (instruction interpretation) – az →utasításaktivizálás második fázisa, amelyben a végrehajtandó →gépi utasítás az →utasításlehívás nyomán már az utasításregiszterben található. Az →utasításdekódoló felismeri a műveleti kódot, és előállítja az ennek megfelelő vezérlőjeleket. Az utasítás →címrészének értelmezése során a már ismert művelet operandusai válnak ismertekké (a →címezhetőség elve alapján az →operatív memóriában közvetlenül hozzáférhetőkké). Az →utasításszámláló regiszterbe bekerül a következő utasítás címe, és megkezdődik az aktuális utasítás tényleges végrehajtása.

utasításkészlet (instruction set) – egy konkrét számítógép →központi vezérlőegysége által felismerhető és végrehajtható →gépi utasítások összessége. A hozzá tartozó utasítások – funkciójuk alapján – több osztályba sorolhatók:

1) az aritmetikai utasítások a négy alapműveletet végzik el; egy gépen belül is többféle változatuk lehet, az ábrázolási módtól függően (fixpontos, lebegőpontos és decimális aritmetikai utasítások);

2) a vezérlésátadó (ugró) utasítások a programban szereplő szekvenciális sorrendhez viszonyítva a végrehajtási sorrendet változtatják meg; függhetnek bizonyos feltételek teljesülésétől, de lehetnek feltételtől függetlenek is;

3) a logikai utasítások speciális logikai műveleteket (→negáció, →logikai szorzás, →logikai összeadás stb.), különböző típusú →léptetéseket, szelekciókat, →maszkolást stb. végeznek;

4) az adatmozgató utasítások az →operatív memória különböző rekeszei és/vagy a →regiszterek közötti adatcserét végzik;

5) a perifériakezelő utasítások az operatív memória és a periferikus egységek közötti

adatcsere lebonyolítására és ellenőrzésére szakosított műveletek.

utasításkód (instruction code) – a →gépi utasításban részt vevő bináris vektor, amelynek értelmezése útján (→utasításdekódoló) a →központi vezérlőegység az ~nak megfelelő műveletet elvégzi. Ismeretlen ~ megjelenésére →megszakítás (→csapda) következik be, amely automatikusan a →felügyelőprogramra adja a vezérlést.

utasításlehívás (instruction fetch) – az →utasításaktivizálás első fázisa, amelyben a →központi vezérlőegység →utasításszámláló regiszterében található →cím átkerül az →operatív memória címregiszterébe, az itt található információ (utasítás) átkerül az átmenőregiszterbe, majd az átmenőregiszterből beíródik a vezérlőegység →utasításregiszterébe (a következő fázis az →utasításértelmezés).

utasításregiszter (instruction register) – a →központi vezérlőegység speciális regisztere, ahol az →utasításszámláló regiszter tartalmának megfelelő →című utasításszó, vagyis a végrehajtandó →gépi utasítás található. Ennek alapján hajtják végre az →utasításértelmezést.

utasításszámláló regiszter (instruction/program counter) – 1. A →központi vezérlőegység speciális regisztere, amely lényegében modulo-N számláló, ahol N az →operatív memória rekeszeinek száma. Az ~ tartalmazza a program végrehajtása közben a mindenkori következő végrehajtandó →gépi utasítás memóriabeli címét, és rendszerint ennek a címnek az önműködő változtatásával vezérlik a végrehajtandó utasítások sorrendjét (szekvenciális végrehajtásnál mindig az aktuális utasítás hosszával növelik az ~ tartalmát, →ugró utasításnál viszont az aktuális utasítás →címrészéből leképzett →abszolút cím kerül az ~be). – 2. →Fordítóprogramok →kódgenerálási fázisában használt speciális adatmező. Aktuális értékét minden egyes →gépi utasítás generálása után az előállított kód hosszúságával növelik, s így válik lehetségessé a →címkékhez és operandusokhoz rendelt →relatív cím meghatározása.

utasítás-végrehajtás (instruction execution) – mindazoknak az elemi műveleteknek a végrehajtása, amelyeket a →központi vezérlőegység az →operatív memóriából kiolvasott →gépi utasítással definiált művelet megvalósítására előír. Rendszerint a következő elemi műveleteket foglalja magában: az utasítások lehívása és értelmezése, az utasításban részt vevő operandusok →címének felismerése és az általa meghatározott memóriarekeszek tartalmának lehívása (idáig tart az →utasításaktivizálás), az előírt műveletet megvalósító mikroműveletek elvégzése a központi vezérlőegység belső regisztereiben, az eredmény elhelyezése a gépi utasítás által erre a célra meghatározott memóriarekeszbe v. regiszterbe.

utófeldolgozás (post-processing) – valamely programtermék kimenő adatain végzett járulékos műveletek összefoglaló megnevezése (ismertebb műveletek: a szűrés, a formázás, az értelmezés stb.). Jellegzetes ~t igényel pl. ma még a →gépi fordítás által előállított nyersfordítás.

utóindexelés (post-indexing) – a →címszámítás egyik módszere: az →indexregiszter tartalma az indirekt címképzés után jön számításba (szemben az →előindexeléssel).

utólapozás (post-paging) – →lapkezelési mód, amelynek célja a tétlenségi idő csökkentése, a kellő méretű szabad laptartalék fenntartása. Lényege az, hogy egyes lapokat kiírnak a →háttértárba, még mielőtt az általuk elfoglalt zónára ténylegesen szükség lenne.

Ü

ütemezési algoritmus (scheduling algorithm) – az az eljárás, amelynek alapján a számítógéprendszer valamely →erőforrásával gazdálkodnak. Célja az erőforrások minél gazdaságosabb kihasználása többfelhasználós környezetben. Az ~ jellegét befolyásolhatja az erőforrás típusa (→memóriagazdálkodás, →időosztásos rendszer stb.). Két alapvető módszere: Az elsőnél a kiosztott erőforrás mindaddig lefoglalt marad, amíg azt ténylegesen fel nem használták, a másodiknál a program elvesztheti az erőforrást, pl. az →időosztásos rendszernél.

ütemezett karbantartás (scheduled maintenance) – pontos határidőkkel tervezett, periodikus →megelőző karbantartás.

ütemező (scheduler) – 1. A →felügyelőprogram egyik igen fontos része, amelynek alapvető funkciója a →multiprogramozás megvalósítása, vagyis a végrehajtásra kész, de várakozó állapotban lévő programok (→feladatok) kiválasztása és indítása. A végrehajtandó →munka kiválasztása →prioritásos, →szekvenciális v. →időosztásos alapon történhet, az erre használt algoritmusok rendszerenként változnak. A kiválasztott munka indítása gyakorlatilag azt jelenti, hogy a vezérlést az utolsó →megszakítás helyét követő utasításnak kell átadni. – 2. →Elosztott vezérlésnél a központi számítógép birtoklásáért vetélkedő programok közötti koordinációt megteremtő rendszerprogram.

üzenet (message) – 1. A kommunikáció lebonyolításának legfontosabb eleme. Az információt az adatátvitelhez fel kell bontani, s ennek a felbontásnak és átvitelnek a legnagyobb, önálló logikai egysége az ~. Az átvinni kívánt hasznos információt minden ~ben vezérlési és értelmezési feladatokat ellátó járulékos adatokkal látják el (→aszinkron adatátvitel, →szinkron adatátvitel stb.). – 2. Az →operációs rendszer által kibocsátott, a →konzolírógépen v. a standard →kimeneti egységen megjelenő szöveges v. kódolt információ, amely az éppen végrehajtás alatt lévő program állapotáról, esetleges erőforrásigényéről, v. a rendszerben felmerülő különböző eseményekről értesíti a gépkezelőt v. a felhasználót.

üzenetcsomag (message packet) – →adatátviteli hálózatokban használatos technológiai átviteli egység, amely az →üzenetek tördelése útján jön létre. A fix hosszúságú ~ irányítása a →fejléc alapján történik, amelyben a küldő, ill. fogadó csomópontra, a bejárandó útra, az ~ sorszámára, a hozzárendelt logikai csatornára stb. vonatkozó információkat rögzítik, s ezt követi a hasznos információ. Strukturális szempontból az ~ is üzenet, de sokkal rövidebb, s olyan logikai csatornán kerül átvitelre, amely – fizikailag változva – az átviteli időt jelentősen lecsökkenti. Egy-egy logikai csatornát természetesen a teljes üzenet átvitelének idejére fenn kell tartani, különben az összetartozó ~ok nem lennének azonosíthatók (szemben az →adatcsomagokkal).

üzenetkapcsolás (message switching) – az →adatállomások közötti összeköttetés olyan megvalósítása, amikor az átvinni kívánt információt v. annak egymás után következő részhalmazait először összegyűjtik, átmenetileg tárolják, s csupán később továbbítják. Ilyenkor az adatcserében részt vevő adatállomások között közvetlen fizikai kapcsolat nem jön létre. Az ~ célja az adatvonalak minél jobb kihasználása: az üzenetkapcsoló központ a vonalak maximális →adatátviteli sebességével adhatja tovább a viszonylag lassú adatállomásoktól összegyűjtött üzeneteket.

üzenetkapcsoló hálózat (message switching network) – olyan →adatátviteli hálózat, amelyben alapvetően nem feldolgozás, hanem csak az →adatállomások által szigorú →protokollok szerint összeállított →üzenetek továbbítása történik. Járulékos feldolgozásként az üzenetek ellenőrzését, ismétlését, felújítását, naplózását, esetleg statisztikai feldolgozását valósíthatja meg.

Számítógépes ~ban átmeneti üzenettárolásra részben az üzenetkapcsoló központi számítógép →operatív memóriáját, részben annak →háttértárát használják. A beérkezett üzenetek az operatív memória →puffereibe, majd onnan a gyors, közvetlen hozzáférésű mágneslemezes háttértárba kerülnek. Az üzenetfeldolgozó program a rendeltetési helyek és a →prioritások alapján →várakozási sorokba rendezi az üzeneteket, amelyek megfelelő időpontokban az operatív memória kimeneti puffereibe, majd onnan az adatvo-

nalakra kerülnek. A számítógépes kapcsolóközpont csak az általa hibátlannak talált üzeneteket fogadja el és továbbítja, átviteli hiba észlelésekor →automatikus ismétléskéréssel próbál korrigálni.

üzenetkoncentrátor (message concentrator) – →üzenetkapcsoló hálózatokban részt vevő kisszámítógép, amely egyszerű →adatállomások egy csoportja részére az →üzeneteket kezeli (összegyűjti, ellenőrzi, esetleg átkódolja, sorbaállítja stb.).

üzenet-sorbaállítás (message queueing) – az →adatátviteli szoftver egyik alapvető feladata; célja a feldolgozó központot elérő és elhagyó →üzenetek →várakozási sorokba rendezése. Az ~ a feldolgozó egységek terhelésétől, az adatátviteli hálózat állapotától és az üzenetek sürgősségétől, →prioritásától függ. Feldolgozás, ill. továbbítás előtt igen gyakran át kell kódolni az üzeneteket, s ez egyúttal a várakozási sorok átrendezését is jelenti.

üzenetszerkesztés (message editing) – az →adatátviteli szoftver egyik alapvető feladata. Célja a beérkező →üzeneteknek a rendszer szempontjából történő uniformizálása, a kimenő üzeneteknek pedig a rendeltetési helyek igényei szerinti előállítása, átalakítása. Tipikus ~i feladat pl. az információk tömörítése, ill. lazítása, az üzenetek átcsoportosítása, a →fejlécek, →vezérlőkarakterek elemzése, beillesztése, elhagyása stb.

üzenetvége karakter (end-of message character) – →üzenetkapcsolásos hálózatokban használt →vezérlőkarakter az átvitelre kijelölt hasznos információ lezárására.

V

VAGY (OR) – kétargumentumú logikai függvény, amely akkor és csakis akkor hamis, ha mind a két argumentum egyszerre hamis. Igazságtáblázata:

VAGY	0	1
0	0	1
1	1	1

válaszábécé (response alphabet) – →kimenő ábécé

válaszidő (response time) – 1. A gépkezelő által végzett utolsó billentyűkezelés és a számítógépes válasz első karakterének vétele között eltelt idő. Az →adatátviteli hálózatok vonatkozásában az az idő, amely a forrásnál (adónál) az üzenetbemenet befejezése és a rendeltetési helynél (vevőnél) az üzenetkimenet kezdete között eltelik. – 2. Számítógéprendszerek teljesítőképességének egyik fontos jellemzője. Arra az időre vonatkozik, amely a munka beadása és az eredmény átvétele között eltelik. (Ez az idő a felhasználó szempontjából értendő, ugyanis hiába hajtódik végre egy program mondjuk egy óra alatt, ha a felhasználó csak másnap kapja kézhez az eredményeket; ez jellemző pl. a →kötegelt feldolgozásra.) A teljesítőképesség másik két alapvető jellemzője az →áteresztőképesség és a →rendelkezésre állás.

A ~ nagyságrendje és fontossága szempontjából a számítógépes rendszerek három csoportba oszthatók:

1) azonos idejű rendszereknél szigorúan betartott felső időkorláton belül kell a számítógépnek kezelnie a bemeneten jelentkező adatokat és kiadnia az eredményeket; ez a felső időkorlát változó nagyságrendű, az illető alkalmazásnak alapvető jellemzője, túllépése teljesen érvényteleníthet az eredmények felhasználhatóságát;

2) kváziazonos idejű rendszereknél a számítógépes válasz késése inkább csak bosszantó, de nem okoz különösebb károkat (ez jellemzi elsősorban a →párbeszédes üzemmódban dolgozó rendszereket);

3) kötegelt feldolgozású rendszereknél a válaszidő nagyságának elég tág határok között nincs különösebb jelentősége, így a munkák ütemezésénél az →erőforrások legjobb hatásfokú kihasználása az elsőrendű cél.

valódi cím (real address) – az →abszolút címmel egyenértékű kifejezés, azzal a különbséggel, hogy a ~ megnevezést olyan címre vonatkoztatva szokták használni, amely →címszámítás eredménye.

valós [adat]típus (real [data] type) – a magas szintű →programozási nyelvek többségében (pl. →FORTRAN, →PL/1, →COBOL, →Pascal stb.) definiált →[adat]típus: értékei a valós számok halmazából kerülnek ki. Számosságuknak csak a számítógépek véges ábrázolási kapacitása szab határt. Minden implementációban van egy olyan korlát, amelynél nagyobb, ill. kisebb abszolút értékű szám az adott gépen már nem ábrázolható. A ~hoz rendelt műveletek az aritmetikai, az összehasonlítási és az értékadó műveletek.

valós idejű rendszer (real-time system) – egy vezérlő és egy v. több vezérelt berendezés →on-line rendszere →valós idejű üzemmódban. Legjellegzetesebb alkalmazási területe a →folyamatvezérlés, ahol általában rövid →válaszidő biztosítása szükséges. A ~ben dolgozó vezérlő számítógéppel szemben támasztott alapvető követelmény, hogy tartósan v. bizonyos meghatározott időpontokban működőképes legyen. Ehhez különleges óvintézkedéseket kell foganatosítani, hogy az esetleg kieső részeket tartalék egységekkel lehessen helyettesíteni (→tartalék redundancia).

valós idejű üzemmód (real-time operation) – számítógéprendszerek olyan üzemmódja, amelyben a megoldandó feladatra jellemző, rendszerint iden rövid →válaszidőt kell biztosítani. A bemenő adatok meghatározott időben jönnek létre, vagyis diszkrét időpontokban érkeznek a számítógépbe, és a feldolgozás eredményeként létrejövő vezérlőjeleket előírt időintervallumon belül kell szolgáltatni. A ~ alapvető jellemzője, hogy a vezérlőjeleket kiadó és az azokat fogadó vezérelt berendezések egymáshoz képest alig eltérő →szinkronban vannak. (Szinonimája az azonos idejű üzemmód.)

valós konstans (real constant) – előjeles v. előjel nélküli, tizedespontot tartalmazó →numerikus konstans. A magas szintű →programozási nyelvekben v. a matematiká-

ból ismert formában (tizedesponttal elválasztott decimális számjegyek sorozata) v. egy →mantissza és egy kitevő formájában használják (pl. 0.275E+4). Függetlenül a →forrásnyelvű formától, a ~ belső kódolására a →lebegőpontos számábrázolást használják.

valós szám (real number) – →programozási nyelvekben a nem egész számok elnevezése. Gépi kódolása általában a →lebegőpontos számábrázolás. A ~ lényegében a véges számú jegyből álló tizedes tört.

változás-[adat]állomány (change file) – →kötegelt feldolgozásnál használatos →[adat]állomány; azokat a →logikai rekordokat tartalmazza, amelyek segítségével a →törzs[adat]állományt naprakész állapotra lehet hozni. Rendszerint →szekvenciális [adat]állomány.

változó (variable) – magas szintű →programozási nyelvekben használatos →szintaktikai kategória: olyan operandus a műveletekben, amelynek értéke a feldolgozás alatt változhat (szemben a →konstanssal). A ~hoz való hozzáférés a program szintjén az →azonosítón keresztül valósul meg. A ~ által képviselt adat attribútumait (pl. →[adat]típusát, ábrázolási formáját, értelmezési tartományát, esetleges kezdeti értékét, érvényességi tartományát stb.) explicit v. implicit →deklarációk útján határozzák meg.

változó hosszúságú rekord (variable length record) – olyan →rekord, amelynek hossza független az ugyanahhoz az →[adat]állományhoz tartozó más rekord hosszától (vagyis a rekordok hossza változó). A hosszúságot mint járulékos szervezési adatot a mágneses →adathordozón tárolni kell. A rekordok hosszát, ill. az ebből következő blokkhosszúságot ún. rekordleíró szóban tárolják, amelynek szabványos hossza 4 →bájt.

A ~ok használata sok esetben jelentős helymegtakarítást eredményez a mágneses adathordozón, mivel mindig csak a tényleges hasznos információ számára kell helyet fenntartani. Természetesen a program a memóriában mindig a maximális hosszúságnak megfelelő munkaterületet jelöli ki a →logikai rekordok számára.

változó partíciók módszere (variable-partitions method) – a →particionált memóriakezelés módszere, a →rögzített partíciók módszerének továbbfejlesztett változata. Az →operatív memória felosztásának elvét megtartja, de végrehajtását az →operációs rendszer hatáskörébe utalja, amely a →partíciók méretét és számát a feldolgozás alatt álló munkák nagyságához és számához igazítja. A módszer előnye rugalmasságában rejlik: mindig a pillanatnyi terhelésnek megfelelően szabályozza a memória kiosztását. Bizonyos fokig hátránya, hogy biztosítani kell a programok →áthelyezését, s így bonyolultabb memóriakezelést igényel. A partíciók túl gyakori változtatása, ill. a szükséges áthelyezések esetleg elfogadhatalan →többletráfordításhoz vezetnek.

várakozás (waiting) – annak az eseménynek, berendezésnek v. programnak az állapota, amely kifejezte bizonyos →erőforrásra (műveletre) vonatkozó igényét, de tényleges kiszolgálása az erőforrás foglaltsága miatt még nem kezdődött meg.

várakozási idő (waiting time) – a →várakozási stratégiákkal kapcsolatos fogalom: az az időintervallum, amely egy →erőforrásra vonatkozó igény sorbaállítása és a tényleges kiszolgálásának megkezdése között eltelik. Egyes stratégiák célja éppen az átlagos ~ minimalizálása.

várakozási modell (waiting model) – valamely várakozási rendszer olyan formális leírása, amely lehetővé teszi a rendszerbe időben beérkező kérések (igények) leírását, a →várakozási sorok strukturális ábrázolását, valamint a →várakozási stratégiák definiálását a →várakozási idők v. az →osztott erőforrás kihasználásának optimalizálása céljából. Tanulmányozható matematikai eszközökkel és →szimulációval egyaránt.

várakozási rendszer (waiting system) – valamely →erőforrásra vonatkozóan hasonló tulajdonsággal rendelkező igények készenléti halmaza. Ezek az igények →várakozásra kényszerülnek, mert a kiszolgálási egység egyszerre csak egy igényt képes kielégíteni. A készenléti igények kiszolgálási sorrendjét a ~ optimális működését biztosító →várakozási stratégia határozza meg. Jellegzetes ~t alkotnak →multiprogramozásos környezetben a →központi vezérlőegység birtoklásáért versenyző, az →operatív memóriában egyszerre jelen lévő programok.

várakozási sor (waiting queue) – olyan →lineáris adatszerkezet, amelynek csak az utolsó eleme után helyezhető el új adatelem, megvizsgálni és törölni viszont csak a mindenkori első elemet lehetséges. Sokszor FIFO-szerkezetnek (first-in – first-out) is nevezik, éppen arra a tulajdonságára utalva, hogy a ~ba bevitt elemek sorrendjét megőrzi, és az azokhoz való hozzáférést a behelyezés sorrendjének megfelelően teszi lehetővé, ha más módszert nem alkalmaznak (→várakozási stratégia). Jellegzetes felhasználási területe: valamely rendszer adott →erőforrására vonatkozó igények sorbaállítása és kiszolgálása olyan módon, hogy az újabb igények mindig a ~ végére állnak be, és mindig a ~ elején álló (legrégebben várakozó) igényt szolgálják ki először.

várakozási stratégia (waiting strategy) – készenléti igények sorrendjét meghatározó módszer, amely az egész →várakozási rendszer optimális működését biztosítja. A kiszolgálási egység, valamint az igények természetétől függően a működés optimalizálását jelentheti: az átlagos →várakozási idő csökkentését, az egész rendszer →áteresztőképességének növelését, a kiszolgálási egység kihasználtsági fokának növelését stb. A leggyakrabban használt ~k közül:

1) a véletlenszerű kiszolgálásnál semmilyen logikai sorrendet nem állapítanak meg a várakozási igények között, egyetlen cél az erőforrás maximális hatékonyságú kihasználása;

2) az „első be – első ki" módszernél az igényeket beérkezési sorrendben szolgálják ki (→várakozási sor, →FIFO-szerkezet);

3) az „utolsó be – első ki" módszernél mindig az utolsónak beérkezett igényt szolgálják ki először (→verem, →LIFO-szerkezet);

4) a prioritásos módszernél az igényeket fontossági (a hozzájuk rendelt elsőbbség) sorrendjében szolgálják ki, ugyanakkor az éppen folyamatban lévő kiszolgálást bármikor megszakíthatja egy nagyobb prioritású igény jelentkezése;

5) az időosztásos módszernél mindegyik igény meghatározott időszeletre jut hozzá az erőforráshoz; ha teljesen kiszolgálták, kilép a várakozási rendszerből, ha nem, újra beáll a várakozási állapotba.

városi hálózat (metropolitan area network) – egy-egy közigazgatási központ területét lefedő, nagy sebességű adatátvitelt biztosító →nyilvános adathálózat, amely rendelkezik a →helyi hálózatok összes tulajdonságával.

VAX (Virtual Addressing eXtended) – 32 →bites nagyszámítógép-sorozat, a DEC amerikai vállalat egyik legfontosabb →számítógépcsaládja. →Operációs rendszere a VAX/VMS.

védőkulcs (protection key) – járulékos adatként használt bináris vektor, amely tartalmával meghatározza az adott erőforrás felhasználhatóságának módját. A legismertebb ~ az →operatív memóriához történő hozzáférést szabályozza: adott memóriazóna hozzáférhető írás és olvasás, csak írás v. csak olvasás céljára, v. lehet teljesen tiltott terület a felhasználó számára. Bármely hozzáférési művelet, amely konfliktusba kerül az illető zóna felhasználhatósági módjával, →csapdához vezet, és a program végrehajtása megszakad. Csak a ~ használata azt jelentené, hogy nem tesznek különbséget az egyes felhasználók között, ezért a ~ot a →hozzáférésikulcscsal összhangban használják, s így a felhasználó típusától függően biztosítják a különböző erőforrások differenciált rendelkezésre állását.

végberendezés (terminal equipment) – →távvezérelt periféria

véges automata (finite automaton) – olyan →automata, amelynek →bemenő ábécéje, felvehető állapotai, ill. →kimenő ábécéje egyaránt véges halmazok.

végeselemek módszere (finite-element method) – a →számítógépes grafika felhasználásának egyik fontos módszere a szerkezetek szilárdsági és kinetikai-kinematikai elemzésében és tervezésében. Lényege az, hogy a vizsgált szerkezetet nagyszámú, véges méretű, idealizált viselkedésű elemi részre osztják, és ezek elemzése útján vonnak le következtetéseket a teljes szerkezet viselkedéséről. A számítások és az elemzés pontosságának előfeltétele a helyes kiindulási adathalmaz, amely egyrészt a véges elemi részek (sokszögek) adataiból, másrészt a terhelőerők, nyomatékok, kezdeti feszültségek, anyagi tulajdonságok, hőmérsékletek stb.

adataiból áll. A kiszámított feszültségértékeket, deformációkat v. lengésképeket ugyancsak grafikusan jelenítik meg.

A ~nek legfontosabb lépései: a vizsgált objektum felosztása, az elemekhez tartozó interpolációs függvények meghatározása, a határátmenetek figyelembevétele alapján a szerkezetet leíró egyenletek meghatározása, ill. megoldása.

végleges kizárás (deadlock) – párhuzamosan futó konkurens folyamatoknál olyan lehetséges állapot, amikor a folyamatok egy része soha be nem következő eseményre, soha fel nem szabaduló →erőforrásra várakozik, az viszont éppen az előzőek valamelyikére várakozik. Az →osztott erőforrásokkal dolgozó rendszerekben léphet fel akkor, amikor egy folyamat teljesen kisajátít egy olyan erőforrást, amelyre a többi folyamat várakozik. Minden olyan rendszerben felléphet, ahol kevesebb erőforrás áll rendelkezésre, mint amennyit a rendszer folyamatainak adott időpontban felléphető igényei szükségessé tesznek. Az ilyen rendszerekben a ~ automatikus felismerése és kiküszöbölése elsőrendű feladat.

végrehajtási hiba (execution/run-time error) – programok végrehajtása közben felléphető hibatípus. A ~k egy része →szintaktikai hibákra, →szemantikai hibákra, →beolvasási hibákra, ill. →kiírási hibákra vezethető vissza, másik része a programozási →logikai hibák egyenes következménye. A ~k külön csoportját alkotja az egyes →hardver →erőforrások elégtelensége, esetleg teljes hiánya, hibás működése stb. Ez utóbbiak rendszerint a helytelen kezelés következményei. A ~k következményeinek megelőzéséhez megfelelő ellenintézkedésekre van szükség: →újraindítási pontok beiktatására, megfelelő mennyiségű és minőségű →[adat]állomány, ill. →adatbázis-másolat előállítására és készenlétben tartására stb.

végrehajtási idő (execution time) – az az időintervallum, amely a program →betöltésétől annak befejeződéséig a program teljes végrehajtásával eltelik. Meghatározó összetevője a →központi vezérlőegység ideje, amit ténylegesen az adott program végrehajtásával tölt, és nem veszi figyelembe a →többletráfordítás időigényét.

végrehajtási könyvtár (run-time library) – →közvetlenül végrehajtható formájú →szubrutinok gyűjteménye, amelynek feladata egy adott →programozási nyelv speciális műveleteinek végrehajtása. E szubrutinokat a →fordítóprogram automatikusan beépíti a →tárgyprogramba a →fordítás alatt. Az →operációs rendszerekben minden programozási nyelvnek megfelel rendszerint egy-egy ~.

végrehajtható utasítás (executable statement) – magas szintű →programozási nyelvekre jellemző olyan utasítás, amelynek hatása a →tárgyprogram végrehajtása során ténylegesen érvényesül. A programok →fordítása szempontjából a ~nak tényleges tárgynyelvű utasítás(sorozat) felel meg. A legismertebb ~ok: →értékadó utasítás, →ciklusutasítás, →vezérlésátadó utasítás stb.

végső felhasználó (end-user) – az a felhasználó, aki a számítógépes szolgáltatásokat az →alkalmazói szoftvert előállító közvetítő felhasználón keresztül veszi igénybe.

végtelen ciklus (infinite/endless loop) – olyan programciklus, amelyből (programozási hiba miatt) hiányzik az ismétlés befejezését meghatározó feltétel, v. ez a feltétel soha nem teljesülhet (→ciklus, →programozási hiba). Rendszerint csak →gépkezelői beavatkozással szüntethető meg, ami egyben a program végrehajtásának rendellenes befejezését is jelenti.

végzetes hiba (fatal error) – valamely számítógépes művelet közben előforduló hiba, amely lehetetlenné teszi az illető művelet (és egyben a program) folytatását. →Fordításnál ~t jelent pl. a →szimbólumtáblázat túlcsordulása, →betöltésnél a megfelelő méretű →operatív memória hiánya, felhasználói programoknál valamely →erőforrás hiánya v. hibás működése, →végtelen ciklus megjelenése, nem kezelhető →csapdák előállása stb.

vegyes kifejezés (mixed expression) – egyes magas szintű →programozási nyelvek által megengedett olyan aritmetikai kifejezés, amelyben a különböző operandusok másmás →[adat]típust képviselnek. Ilyenkor az eredmény típusára, ill. a műveletek elvégzési sorrendjére külön szintaktikai és szemantikai előírások érvényesek. A megfelelő →fordítóprogramok feladatkörébe tartozik azoknak a →szubrutinoknak a →tárgyprogramba törté-

nő beépítése, amelyek a szükséges →adatkonverziókat megvalósítják.

véletlen hozzáférési módszer (random access method) – →közvetlen hozzáférési módszer

véletlen hozzáférésű memória (random access memory) – →közvetlen hozzáférésű memória

véletlen számok (random numbers) – olyan számsorozatok, ahol a számok sorrendjében semmilyen rendszer nem állapítható meg, és az egyes számok előfordulási gyakorisága azonos. Leggyakrabban egyenletes, Poisson- és normális eloszlású ~ra van szükség. Számítástechnikai szempontból a ~at a sztochasztikus jelenségek statisztikus modellezésében használják (→Monte-Carlo módszer). Előállításukra három alapvető módszer használható: véletlen számtáblázatok, speciális →hardvereszközök (véletlen számgenerátorok), ill. speciális →szoftvereszközök (pszeudovéletlen számgenerátorok).

véletlenszerű adatútképzés (random routing) – olyan →adatútképzés, amelynél az →adatcsomagot a fogadó csomópont véletlenszerűen kiválasztott összeköttetésen küldi tovább. Az összeköttetések teljesen véletlenszerű megválasztása helyett a valószínűség valamilyen fajta kiválasztásával lehet elérni azt, hogy az adatcsomagokat jó közelítéssel a helyes irányba vezessék. Ennél a módszernél azonban az összeköttetés- v. csomóponthibák kiküszöbölésére nincs lehetőség.

verem (stack) – olyan →lineáris adatszerkezet, amelynek csak a mindenkori utolsó elemével végezhetők műveletek (megvizsgálás, ill. törlés), és új elem mindig csak az utolsó elem után helyezhető el. Sokszor LIFO-szerkezetnek (last-in – first-out) is nevezik, éppen arra a tulajdonságra utalva, hogy az utolsó bevitt adatelem lesz az adatszerkezet egyetlen elérhető eleme. A ~ egyik legfontosabb előnye, hogy a felhasználónak nem kell foglalkoznia a memóriacímekkel, s csak két specifikus utasításra van szüksége az adatszerkezet kezeléséhez: az új adatelem elhelyezésére (PUSH) és az utolsó adatelem kiolvasására, ill. törlésére (POP).

Számos gyakorlati alkalmazására való tekintettel sok számítógépen a ~szerkezet →hardverúton is megvalósított, s egyes magas szintű →programozási nyelvek is lehetővé teszik a ~ típusú adatszerkezet definiálását.

veremmutató (stack pointer) – →verem típusú adatszerkezetek szervezésénél használt járulékos adat, amely mindig a verem utolsó (tehát elérhető) adatelemének →címét tartalmazza. →Veremorientált számítógépeknél a ~ szerepét speciális →regiszter látja el, amelynek mozgatását (aktualizálását) a →központi vezérlőegység végzi: minden új adatelem rögzítésével a ~ tartalma az illető elem hosszával nő, kiolvasásnál (törlésnél) pedig ugyanannyival csökken.

veremművelet (stack operation) – →verem típusú adatszerkezetek kezelésével kapcsolatos szervezési művelet. Alapjában véve két ilyen művelet van: az új adatelem elhelyezése (rögzítése) az adatszerkezetben (PUSH) és a legutolsó adatelem kiolvasása (törlése) onnan (POP). Mindkét művelet végső soron a →veremmutatóval dolgozik: beírásnál az adatelem hosszával növeli, kiolvasásnál ugyanazzal az értékkel csökkenti annak aktuális értékét. →Veremorientált számítógépeknél az →utasításkészlethez tartozik ez a két speciális művelet is.

veremorientált számítógép (stack-oriented computer) – olyan számítógépes →architektúra, amely elektronikus (→hardver) úton valósítja meg a →verem típusú adatszerkezet kezelését. Elsősorban a →mikroprocesszoros architektúrákra jellemző, de kisszámítógépeknél is gyakran megtalálható. A verem lehet belső, a mikroprocesszorba beépített és korlátozott mélységű, és lehet külső, az →operatív memóriába kihelyezett is. Ez utóbbinál a felhasználó határozhatja meg a verem mélységét és memórián belüli helyét. Mindkét esetben a →veremmutató a mikroprocesszor belső regisztere.

versenyhelyzet (contention) – több folyamat, berendezés, program olyan állapota, amelyben a közösen használt erőforrásra egyszerre jelentenek be igényt. ~ alakulhat ki pl. a →központi vezérlőegység birtoklásáért, adatállomások között a →félduplex csatorna használatáért stb. →Szoftver erőforrásokkal kapcsolatban is előállhat (azzal a megjegyzéssel, hogy a →tiszta kódolású programok „egyszerre" több igénylőt is kiszolgálhatnak).

A ~ felismerése és megoldása (az igénylők valamilyen szempont szerinti ütemezése) az →operációs rendszer legfontosabb feladatai közé tartozik (→konkurens elvű rendszer).

verziószám (version number) – kétjegyű decimális szám, amely lehetővé teszi egy →[adat]állomány azonosítását két v. több azonos nevű és azonos →generációszámú [adat]állomány között. Szabványos helye a HDR1 típusú →[adat]állomány-kezdőcímkében van.

vezérjelgyűrű (token ring) – hurkolt topológiájú →adatátviteli hálózatok vezérlésére használt módszer. Ennél a kiépítésnél az adatforgalom mindig egy jól meghatározott irányban bonyolódik, tehát a hálózatra felfűzött →csomópontok is jól meghatározott sorrendben kerülhetnek aktív állapotba, ezt az állapotváltozást pedig egy rövid bitsorozatból álló vezérjel váltja ki. A vezérjel szekvenciálisan járja körül az összes →adatállomást (kering a gyűrűben). Mihelyt az adott adatállomás a vezérjel birtokába jutott, aktív állapotba kerül: veszi az előző állomás által küldött adatokat, továbbítja a nem neki szóló adatokat a következő állomás felé, majd a vezérjelet szintén továbbadja a soron következő állomásnak.

vezérkártya (control card) – →kötegelt feldolgozású →operációs rendszereknél a →munkavezérlő nyelv utasításainak hordozója, amely a végrehajtandó →munkára vonatkozó információkat tartalmazza: a memóriaigényt, a felbecsült végrehajtási időt, a →prioritást, a könyvelési és számlázási adatokat, a →szoftver, ill. →hardver erőforrások igénylését, a →rendszeprogramok, ill. a →kezelőprogramok működéséhez szükséges paramétereket stb. Felismerése és feldolgozása a →munkaütemező feladata.

vezérlésátadó utasítás (control statement) – magas szintű →programozási nyelvek olyan típusú utasítása, amely lehetővé teszi az egyébként szekvenciális struktúrájú program különböző részeinek logikai sorrendben történő végrehajtását. A ~ lehet feltételes v. feltétlen →ugró utasítás (ez máskülönben az →asszembler nyelvekben is megtalálható), →cilklusutasítás, →eljáráshívó utasítás stb.

vezérlési információ (control information) – adatfeldolgozásban használt olyan járulékos információ, adat, amely nem vesz részt ténylegesen a feldolgozásban, de jelenléte nélkülözhetetlen a folyamatok végrehajtása szempontjából. Több kategóriája ismeretes: állapotinformáció (→programállapotszó, →csatorna-állapotszó stb.), szervezési információ (→[adat]állományok, ill. →adatbázisok szervezése, →lapkezelés, →virtuális memória stb.), adatátviteli információ (→szinkronizálási információk, →hibakorlátozó kódok, →vezérlőkarakterek stb.).

vezérlő (controller) – számítógépes →architektúrában valamely részegység vezérlését ellátó berendezés, ill. a számítógép és a hozzákapcsolt eszköz közötti információ áramlását irányító, kiszolgáló áramkörök összessége (pl. →programozható megszakításvezérlő, →programozható perifériaillesztő, →képernyőadapter, →DMA-vezérlő, →multiplexcsatorna, →szelektorcsatorna stb.

vezérlőblokk (command block) – a fizikai →b/k műveletek elvégzéséhez szükséges információkat magában foglaló bitvektor. A felhasználó ~ban helyezi el az adatátviteli igényét meghatározó információkat, l. a →csatornaprogram címe, az igényelt →periferikus egység logikai címe, az adatátvitel idején bekövetkező esetleges hibákat feldolgozó eljárás címe stb. Ugyancsak a ~ban kaphat a felhasználó olyan információkat, amelyek a b/k művelet befejezésére, ill. az átvitel helyességére vonatkoznak (→adatvezérlő blokk, →vonalvezérlő blokk).

vezérlőegység (control unit) – a →processzor funkcionális alegysége, amely adatátviteli és/vagy aritmetikai és logikai műveletek végrehajtásához szükséges vezérlőjeleket állít elő. Két alapvető típusa a huzalozott vezérlés, ill. a →mikroprogramozott vezérlés elve alapján működik (→központi vezérlőegység).

vezérlőkarakter (control character) – →adatátviteli hálózatokban használt járulékos információ, amely lehetővé teszi a hasznos információ szabványos szervezését, valamint a különböző adatátviteli módszerek (→szinkron adatátvitel, →aszinkron adatátvitel) megvalósítását.

vezérlőmodul (control module) – a →moduláris programozás elvei alapján előállított program olyan aktív modulja, amely az egész program logikai szerkezetének hordozója. Fő

funkciója az →eljárásmodulok és a →b/k modulok előírt sorrendben történő aktivizálása, az egész program működésének irányítása. Bár fő sajátossága az, hogy nem végez tényleges feldolgozást, tartalmazhat olyan tevékenységeket is, amelyeket nem érdemes külön eljárásmodulban elhelyezni. Minden modulárisan előállított programnak tartalmaznia kell legalább egy ~t, de lehet egyszerre több ~ja is, a különböző logikai szinteknek megfelelően.

vezérlőpult (operator's console/keyboard) – számítógéprendszernek az a →hardverberendezése, amelynek útján az →operációs rendszer és a gépkezelő közötti információcsere megvalósul. A gépkezelő kapcsolók, nyomógombok és/vagy alfanumerikus billentyűzet útján vihet be a rendszerbe információkat, és speciális →nyomtató és/vagy katódsugárcsöves →megjelenítő útján juthat információkhoz (→konzolírógép, →terminál). A ~ kiemelt jelentőségű funkciója a →gépkezelői beavatkozás, ami lényegében a legmagasabb →prioritással rendelkező →megszakítás és a rendszerrel folytatandó dialógus kezdeményezésére szolgál.

vezérlőszerkezet (control structure) – a →lépésenkénti finomítás elve alapján megvalósított programok résztevékenységeit ütemező szerkezet. Statikus megközelítésben a →forrásprogram nyelvi szerkezeteiben, dinamikus szempontból a program végrehajtásának dinamizmusában, a résztevékenységek egymásutániságában nyilvánul meg. Három alapvető típusa van. A szekvencia jól meghatározott résztevékenységek sorozatát jelenti. A szelekció két v. több egyenlő esélyű résztevékenységet jelöl, amelyek közül azonban csak egy kerülhet végrehajtásra. Ennek kiválasztása mindig egy feltétel (v. szelektor) kiértékelése útján történik. Az iteráció egy bizonyos résztevékenységnek egymás után többszöri végrehajtását jelenti. Ha a végrehajtások száma előre ismert, akkor a kilépést egy egyszerű számláló biztosíthatja, ellenkező esetben valamilyen kilépési feltétel szükséges, amelyet minden egyes ciklus előtt v. után ki kell értékelni.

vezérszalag (command tape) – perforált papírra író →nyomtatóknál használt lyukasztott, zárt papírszalag, amelynek segítségével a nyomtatási papír továbbítását (új lapbeállítást, soremelést stb.) vezérelni lehet.

videolemez (video disk) – →kompaktlemez

villogás (blinking) – →képernyőn megjelenített →képpontok jellemzője, amelynek lényege, hogy az illető képpont rendszeres időközönként, megszakításokkal jelenik meg. A váltakozó képek jó minőségű megjelenítése érdekében a megjelenítő áramkör ciklikusan vetíti a képernyőtár tartalmát a képernyőre, ami már maga is ~t eredményez. Ennek kiküszöbölésére a →képfrissítés frekvenciája legalább 50-60 Hz kell hogy legyen. ~t szándékosan is elő lehet idézni (ez 1-10 Hz között nem bántó) a képernyő egyes részeinek kiemelése érdekében (ez a tulajdonság a képpont fényessége és színe mellett kódolható).

virtuális adatállomás (virtual terminal station) – olyan ideális →adatállomás, amelyet a tényleges adatállomás-változatok egységes kezelése céljából definiáltak. Jellegzetessége, hogy rendelkezik mindazokkal az adottságokkal, amelyeket bármelyik felhasználó igényelhet. A felhasználói adatállomás-kezelő program leképzési eljárással felelteti meg a ~t a tényleges, valódi adatállomásnak. Így elérhető, hogy két felhasználó úgy kommunikáljon egymással, mintha mindkettő ~sal rendelkezne, függetlenül a tényleges berendezésektől (ezek utólag aránylag könnyen cserélhetők újabb típusokra is, anélkül hogy ez érdemlegesen befolyásolná a kommunikációs eljárásokat).

virtuális adatelem (virtual data item) – olyan adatelem, amely nem található meg ténylegesen a →fizikai adatbázisban, de szerepel a logikai leírásban (→adatbázissémában), s így mégis kezelhető az →adatbázis-kezelő rendszer segítségével. A ~ v. →származtatott adatelem, v. egy külső →[adat]állományból beolvasott adat.

virtuális cím (virtual address) – a programok →logikai címtartományához tartozó olyan →cím, amely nem veszi figyelembe a számítógépek →operatív címtartományának véges (fizikai) kapacitását. Az →operációs rendszer feladatai közé tartozik, hogy a logikai címeket fizikai (→abszolút) címekké átalakítsa. Szerkez ·ti szempontból a ~ v. kétkomponensű (c=/s,w/, ahol s a →szegmens azo-

nosítója, v pedig a szegmensen belüli →relatív cím), v. háromkomponensű logikai cím (c=/s,p,w/, ahol s a szegmens azonosítója, p a szegmenshez tartozó →lap száma, w pedig a lapon belüli relatív cím).

virtuális gép (virtual machine) – 1. Számítógépek, →programozási nyelvek, algoritmusok stb. elméleti tanulmányozására előállított matematikai modell, formális rendszer. Alapvető jelentősége az, hogy matematikailag megalapozott eljárásokat, absztrakciókat használhat a jelenségek tanulmányozására, s ugyanakkor bármely – a gyakorlatban előfordulható konkrét – számítógéprendszer elvonatkoztatott tulajdonságait figyelembe veszi. – 2. Az →elosztott vezérlés körülményei között dolgozó →számítógép-hálózatokkal kapcsolatos fogalom. Azt az absztrakt számítógépet jelöli, amelynek vezérlése – az →operációs rendszer szempontjából – egyenértékű az egész hálózat vezérlésével.

virtuális memória (virtual memory) – az a →logikai címtartomány, amelyet egy program minden megszorítás nélkül használhat, s amely sokkal nagyobb lehet, mint a rendelkezésre álló →operatív címtartomány. Az →átfedési szerkezet használata már lehetővé tette a felhasználók számára, hogy programjaik tördelésével megosszák ezeket az →operatív memória és a →háttértár között. Ennek az elvnek a továbbfejlesztése a ~, amelynek lényege az, hogy az →operációs rendszer úgy teszi lehetővé a kétszintű memória felhasználását, mintha az egyszintű, nagyméretű operatív memória lenne. A programok megfelelő tördeléséről az operációs rendszer gondoskodik, a művelet teljesen transzparens a felhasználók számára (→virtuális memóriacímzés, →lapkezelés, →címátalakítás).

virtuális memóriacímzés (virtual memory addressing) – →virtuális memóriát megvalósító címzési módszer, amelynél a programozó úgy használhatja az →operatív memóriából és a →háttértárból álló kétszintes hierarchiát, mintha az egyszintű, nagyméretű operatív memória lenne. A →virtuális címeket az →operációs rendszer képezi le →abszolút címekké (→címátalakítás).

A ~ mechanizmusa a következő: a végrehajtandó program megosztva helyezkedik el az operatív memóriában és a háttértárban; a program hivatkozásai virtuális (logikai) címek; minden egyes hivatkozásnál az operációs rendszer értelmezi a logikai címet, és megállapítja, hogy az leképezhető-e az operatív memóriában tartózkodó programrészhez kapcsolt valamely abszolút címre; ha igen, elvégzi az előírt leképzést és előállítja a megfelelő abszolút címet; ha nem, →megszakítást hoz létre, amelynek feldolgozása révén a kívánt programrész bekerül az operatív memóriába (→lapkezelés, →laphelyettesítési algoritmus).

virtuális számítógép (virtual computer) – többfelhasználós környezetben – elsősorban →időosztásos üzemmódban – az a látszólagos erőforrásrendszer, amely látszólag kizárólagosan egy-egy felhasználó rendelkezésére áll. E rendszer →hardverelemei virtuális, azaz látszólagos elemek, és eltérnek azoktól a reális, tényleges elemektől, amelyek a felhasználó programjának végrehajtásában részt vesznek. A virtuális és reális elemeket hozzárendelési v. leképzési rendszer kapcsolja össze (pl. →virtuális memória, →virtuális adatállomás stb.).

virtuális távközlési hozzáférési módszer (virtual telecommunications access method) – az IBM System/370 számítógéprendszer legfontosabb →szoftverterméke az →elosztott feldolgozás, a →számítógép-hálózatok környezetében. Alapvető feladata a felhasználói programok és a →terminálok közötti kommunikáció vezérlése, így az →interfész szerepét tölti be a számítógép-hálózat és felhasználói között.

virtuális terminál (virtual terminal) – →virtuális adatállomás

vírus (virus) – olyan speciális önreprodukáló program, amely más programtermékekbe beépülve a legkülönbözőbb mellékhatásokat idézheti elő. Hatása az egyszerű, ártatlan csínytevéstől a legkomolyabb számítógépes bűntényig terjedhet, ugyanis igen komoly károkat okozhat készakarva az alkalmazói →[adat]állományokban, ill. →adatbázisokban. Elnevezése igen találó, mert terjedése éppen olyan ellenőrizhetetlen, mint természetes névadójáé.

visszaállítási másolat (back-up copy) – →operációs rendszerek megbízhatóságát növelő eszköz. Lényegében az operációs

rendszer – közvetlenül →rendszergenerálás utáni – változatának tárolását jelenti valamilyen mágneses adathordozón (rendszerint mágneslemezen v. mágnesszalagon). Minden olyan esetben, amikor a rendszer valamilyen végzetes hiba folytán holtpontra (blokkolt állapotba) kerül, a ~ról egyszerű másolással visszaállítható az operációs rendszer eredeti állapota, s így nem szükséges megismételni a rendszergenerálás sokszor elég hosszadalmas műveletét.

visszafelé olvasás (back reading) – A →mágnesszalag →soros hozzáférésű adathordozó, tehát rendszerint a szalag elejétől szekvenciálisan olvassák végig. A legtöbb rendszer viszont lehetővé teszi, hogy a mágnesszalagot visszatekercselés nélkül, de ugyancsak szekvenciálisan ~sal is fel lehessen dolgozni. Ez igen hasznos olyan munkáknál, ahol sűrűn kell az előállított →[adat]állományt rögtön visszaolvasni: ~sal megtakarítható a →visszatekercseléssel elvesztett idő. Szervezési szempontból a ~ra az ad lehetőséget, hogy az →[adat]állomány-zárócímkék tartalmazzák mindazokat az információkat, amelyekre az →[adat]állomány-kezelő rendszernek szüksége van.

visszafogott kiértékelés (lazy evaluation) – egyes műveletek végrehajtási módja. Lényege az, hogy az illető művelet végrehajtása csak akkor és csak olyan mértékben halad előre, amikor és amennyire az feltétlenül szükséges. Klasszikus példája a ~nek az a mód, ahogyan egyes →fordítóprogramok a →logikai kifejezéseket kezelik. Ezeknél a kiértékelés csak addig tart, amíg nyilvánvalóvá nem válik a kifejezés értéke (a további műveletek hatásától függetlenül). A módszert sok →segédprogram is alkalmazza: pl. az →[adat]állományokat összehasonlító segédprogram csak az első különbség felfedezéséig folytatja a műveletet.

visszagörgetés (roll back) – a meghibásodott →adatbázis helyreállításának egyik módszere, amely az érintett →rekordok aktualizálás előtti képét használja fel. Menete a következő: a hiba felfedezése után betöltik az utolsó →adatbázis-másolatot; az adatbázisba visszavezetik azoknak a rekordoknak aktualizálás előtti (tehát helyes) állapotát, amelyekre módosító művelet vonatkozott a hiba bekövetkezése előtt; a helyes →tranzakciókkal ismét végrehajtják az aktualizálásokat.

visszaigazolás (acknowledge) – az →automatikus ismétléskéréssel dolgozó →adatátviteli hálózatokban az adóállomás a helyes vételről ~t igényel (a hibátlan átvitel elismerése). Ellenkező esetben megismétli a hibásnak vélt üzenetet.

visszaigazolás karakter (acknowledge character) – pozitív →visszaigazolás jelzésére használt egyezményes →vezérlőkarakter.

visszalépegetés (backtracking) – a →logikai programozásnak a →mintaillesztés mellett legfontosabb jellegzetessége, módszere. A problémamegoldás előre megfogalmazott és megfelelő módon ábrázolt sémák, minták alapján történik. A konkrét feladat megoldása a rendelkezésre álló sémákkal való sorozatos illesztések útján valósul meg. Minden egyes lépésben, ahol az illesztés zsákutcába jut (vagyis bebizonyosodik, hogy az aktuális lépés már nem vezethet eredményre), ~sel regenerálódik az az előző állapot, amelyben a mintaillesztés még eredményesen végződött, s innen további – az adott feladatra vonatkoztatható – illesztések sorozatos alkalmazásával próbálják a megfelelő sémát kiválasztani. Ezt a „próbálgatásos" módszert folytatják mindaddig, amíg v. megoldódik a feladat, v. kiderül, hogy az adott sémákkal a feladat nem oldható meg.

visszalépés (backspace) – 1. Szekvenciális feldolgozásnál az aktuális információegységet közvetlenül megelőzőre történő visszatérés. →Szekvenciális [adat]állományok feldolgozására a magas szintű →programozási nyelvek speciális utasítással teszik lehetővé a ~ elvégzését (pl. BACKSPACE a →FORTRAN-ban). Bizonyos esetekben a ~ egyben az aktuális információegység automatikus törlését, ill. módosítását is jelentheti. – 2. Billentyűzettel ellátott →terminálokon a képernyőn való ~re speciális billentyű szolgál.

visszametszés (prunning) – →mintaillesztést alkalmazó programozástechnikáknál (pl. →logikai programozás, →heurisztikus programozás) használt gyorsító eljárás. Lényege az, hogy a mintaillesztés előrehaladását erőszakosan megszakítják, ha előreláthatóan nem fog eredményhez vezetni, s →visszalépegetéssel olyan helyzetből folytatják a próbálga-

tást, ahol még van remény megfelelő megoldásra.

visszatekercselés (rewind) – soros hozzáférésű →adathordozókra (elsősorban a →mágnesszalagra) jellemző művelet. Feldolgozás után a mágnesszalag ~ útján állítható újra felhasználható állapotba.

visszatérés (return) – valamely hívott eljárás (→eljáráshívás) végrehajtásának befejezése után a hívó program végrehajtásának folytatása. A ~ rendszerint közvetlenül a hívó utasítás után következő utasításra adja a vezérlést, de egyes →programozási nyelvek lehetővé teszik a hívó program végrehajtásának folytatását más, előre definiált pontról is.

visszatérési cím (return address) – valamely hívó eljárás (→eljáráshívás) olyan pontja (címe), ahol a szekvenciális végrehajtás folytatódik a hívott eljárás végrehajtása után. Rendszerint a hívó utasítást szekvenciálisan követő első utasítás címe. Az eljáráshívás egyik fontos feladata a ~ megőrzése és visszaállítása. Ezt a műveletet a magas szintű →programozási nyelvek →fordítóprogramjai automatikusan végzik el (vagyis megfelelő →gépi utasításokat generálnak a végrehajtására), →asszembler nyelvű programozásnál viszont többnyire a programozó feladata. A →veremorientált számítógépeknél az eljáráshívást rendszerint speciális gépi utasítás hajtja végre, amely automatikusan a ~ megőrzéséről is gondoskodik (a →verembe helyezi, majd →visszatéréskor onnan kiolvasva állítja vissza).

visszatérési kód (return code) – →kötegelt feldolgozásnál a →munkák láncolt végrehajtásában szerepet játszó járulékos szervezési adat, amely az aktuális munkalépés végrehajtását minősíti. Előfordulhat, hogy adott munkalépés nem megfelelő végrehajtása feleslegessé, olykor veszélyessé teszi a további lépéseket. Így pl. végzetes hibával végződő →fordítás után teljesen felesleges a →kapcsolóprogram v. a →betöltőprogram aktivizálása. A →munkaütemező egy-egy újabb lépés aktivizálása előtt mindig megvizsgálja a ~ot, s annak értéke alapján hoz döntést. A →munkavezérlő nyelv parancsaiban a felhasználó maga is feltételeket állapíthat meg a ~ értékére vonatkozólag.

visszhang (echo) – 1. Az információátvitel ellenőrzése céljából a vevő által az adóhoz visszairányított információ. Az →adatátviteli hálózatokban mint →hibakorlátozó stratégiát használják (→visszhangellenőrzés). – 2. →Párbeszédes üzemmódnál olyan parancs, amely egy batch programüzenetet küld működéséről.

visszhangellenőrzés (echo checking) – az az →adatátviteli hálózatokban használatos →hibakorlátozó stratégia, amely nem igényel →hibakorlátozó kódolást. Lényege az, hogy a vevőoldali adatállomás visszaküldi az egész →üzenetet a forrásoldali adatállomáshoz, ahol bitről bitre összehasonlítják őket. A hibásnak talált üzenetet az adóállomás automatikusan újraközvetíti. Ha az ellenőrzés hibátlannak minősül, az adóállomás egyetlen karakterrel utasítja a vevőállomást, hogy az üzenetet küldje végleges rendeltetési helyére.

vonalkapcsolás (line switching) – az →adatállomások közötti összeköttetés olyan megvalósítása, amelynél a bemenővonal közvetlenül kapcsolódik egy kimenővonallal, vagyis az adatcserében részt vevő adatállomások között közvetlen fizikai kapcsolat jön létre. Az átvitel befejezése után a kimenővonal felszabadul, és újabb bemenővonal rendelkezésére állhat.

vonalkapcsolásos hálózat (line switching network) – a hagyományos postai távközlő rendszerekben használt hálózattípus, amelynek legfontosabb jellemzői: az összekapcsolt →adatállomások között a kommunikáció teljes ideje alatt közvetlen a kapcsolat; a kapcsolat előállítása előzetes címzéssel („felhívással") történik; a benne részt vevő szakaszok az összekapcsolás egész ideje alatt foglaltak; a vonalszakaszok összekapcsolása viszonylag hosszú időt igényel.

A ~ok hagyományos elektromechanikus központjait rendre felváltják a számítógéppel vezérelt központok, amelyekben a kapcsolás tisztán elektronikus úton valósul meg. A programvezérlés rendkívül rugalmassá teszi az egész rendszert. A ~ok vezérlési feladatainak megoldására speciális magas szintű →programozási nyelvet is kifejlesztettek (→CHILL).

vonalkód (bar code) – nyomtatott, különböző számú, szélességű és egymáshoz viszonyított távolságú, párhuzamos vonalból álló →kód. Legismertebb felhasználása az ipari termékek azonosítására használt egysé-

ges címke, amelyet speciális olvasóval lehet letapogatni, a nyert információk alapján pedig naprakész raktárkészlet-figyelés, forgalmi statisztikák stb. készíthetők. Legismertebb változatai az →EAN-kód, a 2/5 Code, Codebar, Code alpha 39 stb.

vonalkoncentrálás (line concentration) – →adatátviteli hálózatokban használatos eljárás, amelynél adott számú elemi bemenő →kommunikációs csatorna dinamikusan és igény szerint osztozkodik kevesebb számú kimenőcsatornán. Célja az adatátviteli forgalom egyenletesebbé tétele, s az →üzenetek statisztikus jellegén és sorbaállíthatóságán alapszik. Jellegzetessége, hogy a felmerülő átviteli igények torlódásánál foglalt állapot is előadódhat, amikor az igények csak →várakozással elégíthetők ki.

vonalmultiplexelés (line multiplexing) – →adatátviteli hálózatokban használatos eljárás, amelynek során egy adatvonalat előre meghatározott módszer szerint osztanak fel elemi csatornákra. Mivel minden elemi bemenőcsatornához elemi kimenőcsatorna is tartozik, a ~ nem eredményezhet csatornafoglaltságot, az igények kielégítése →várakozás nélkül történik.

vonalrajzoló kijelző (vector display) – olyan →grafikus kijelző, amelynél a rajzolást végző elektronsugár bármely időben a képernyő tetszőleges pontjába mozgatható, de az ábrákat különböző hosszúságú vonalszakaszok sorozatával állítják elő. Lényeges működésbeli különbsége a →raszteres kijelzővel szemben az, hogy az elektronsugár nem a hagyományos tévékészülékhez hasonló mozgást végez, hanem a kép vonala mentén haladva vektorokból hozza létre az ábrát. Egyes ~k görbe ívek menti eltérítő generátorokkal is el vannak látva, s így simább átmenetű görbe vonalú ábrákat állíthatnak elő (általában másodfokú íveket használnak).

Működési elve alapján a ~ lehet képfrissítő v. tárolócsöves. Az előbbinél ahhoz, hogy a kép állandóan látható maradjon, az elektronsugarat ciklikusan végig kell vezetni a nyomvonalon (ez a tulajdonképpeni képfrissítés).

vonalvezérlő blokk (line control block) – az →adatátviteli szoftver által felhasznált állapotinformációk táblázatos gyűjteménye. A felhasználói programokban minden egyes →kommunikációs csatornához ~ot rendelnek, s ebben rögzítik mindazokat az információkat, amelyek a felhasznált átviteli vonal jellemzőire, a →pufferek méretére és számára, a lekérdezési sorrendre, a kódolásokra, a szerkesztésekre stb. vonatkoznak. Ezek révén és az adatátviteli szoftver segítségével a felhasználók úgy kezelhetik a földrajzilag távoli →terminálokat, mint a számítógéprendszerek klasszikus →periferikus műveit.

V-szabványok (V-series recommandations) – a CCITT ajánlások egyik sorozata, amely a telefonhálózaton lebonyolított adatátvitel →illesztőegységeire (→interfészeire) vonatkozik, elsősorban →vonalkapcsolásos v. →csomagkapcsolásos üzemmódban. Így pl. a V.7 szabvány definiálja a telefonhálózaton lebonyolított adatátvitelben használt fogalmakat, a V.15 definiálja az →akusztikus csatolót, a V.23 szabványosítja a 600/1200 bps-os →modem felhasználását, a V.26 írja le a 2400 bps-os modemet, a V.41 a kódfüggetlen hibaellenőrzési rendszert stb.

W

Warnier-módszer (Warnier's method) – a →strukturált programozás módszertanának egyik gyakorlati változata, amelyet Jean-Dominique Warnier és munkatársai dolgoztak ki a hetvenes évek elején. Központi alapelve az a megfigyelés, hogy egy →program akkor követi a leghatékonyabban a megoldandó feladat szerkezetét, ha az ill. programot a feldolgozandó adatok szerkezetéből származtatják. Ilyen vonatkozásban a →Jackson-módszerrel nagyon sok rokon vonást tartalmazó módszer.

WATFOR (WATerloo FORtran) – a kanadai Waterloo Egyetemen kifejlesztett speciális →fordítóprogram a →FORTRAN programozási nyelv számára, amely azonnal betölthető és végrehajtható →tárgyprogramot állított elő (tehát egyszerre látta el a fordítóprogram, a →kapcsolóprogram és a →betöltőprogram szerepét).

WordStar – a →CP/M →operációs rendszer számára kidolgozott, majd az →MS-DOS alatt is elterjedt egyik legismertebb →szövegszerkesztő, műveletválasztéka szinte szabvány jellegű a számtalan hasonló típusú programtermék számára.

X

XENIX – a Microsoft cég által az IBM →személyi számítógépeire kifejlesztett, →UNIX típusú →operációs rendszer.

X-szabványok (X-series recommendations) – a CCITT ajánlások egyik sorozata, amely a telefonhálózatokon lebonyolított →digitális adatátvitel →protokolljaira vonatkozik, de ajánlást ad számos →modem v. →interfész megvalósítására is. Így pl. az X.1 írja le a →nyilvános adathálózatok által nyújtott szolgáltatásokat, az X.15 definiálja a nyilvános adathálózatok alapvető fogalmait stb.

Z

Z – Konrad Zuse által a két világháború között Németországban épített számítógépek családneve. A Z1 mechanikus, a Z2 elektromechanikus, nem programozható gép volt. Az 1941-ben épített Z3 már programozható →jelfogós számítógép volt.

zárójelmentes jelölés (parenthesis-free notation) – →lengyel forma

zárt gépterem (closed shop) – →kötegelt feldolgozással üzemelő számítóközpont szervezési módja. Jelentése az, hogy a végrehajtandó program tulajdonosa (a számítóközpont bérlője) nem férhet hozzá közvetlenül a számítógéphez. Programját és feldolgozandó adatait átadja a számítóközpontnak, a végrehajtást – géptermi utasítások alapján – a gépkezelő végezteti el. A →nyitott géptermemmel szemben előnye, hogy fegyelmezettebb a munkafolyamat, biztosítható az elvégzendő →munkák optimális sorrendje, és az ügyfeleknek nem szükséges elsajátítaniuk a számítógép kezelését. Sokszor viszont hátrány, hogy előre nem látott eseményeknél a gépkezelő nem tud hatékonyan beavatkozni.

zárt hatásláncú vezérlés (closed-loop control) – a →folyamatvezérlés olyan megvalósítása, amelynél az ellenőrzés és a tényleges vezérlés egységes rendszert alkot, s nem igényel emberi beavatkozást. A →folyamatvezérlő számítógép a vezérelt folyamat állapotváltozásairól a közvetlenül mért adatok feldolgozása révén vesz tudomást, s közvetlenül, emberi beavatkozás nélkül juttatja vissza a folyamathoz a szükséges vezérlőjeleket. A ~ csak →valós idejű üzemmódban valósítható meg, s rendszerint speciális kiépítettségű →architektúrát igényel (megfelelő mérőműszerek, →analóg-digitális, ill. →digitális-analóg konverterek stb.).

zárt szubrutin (closed subroutine) – olyan utasítássorozat, amelynek kódja csak egy példányban van jelen a programban, s minden olyan helyen, ahol ezt a programrészt végre akarják hajtani, speciális hívóutasítással aktivizálják. A →nyílt szubrutinnal szemben itt csak az aktivizáláshoz szükséges utasítások vannak jelen a felhasználás helyén (→eljáráshívás).

zavarérzékelő program (malfunction program) – →operációs rendszerekben általában megtalálható olyan speciális program, amely a számítógéprendszer egyes →hardveregységeinek hibás működését érzékelni tudja, s erről a gépkezelőt értesíti. Korszerűbb változatai egyes zavarok elhárítására is képesek. Így pl. az →operatív memória v. a →periferikus művek periodikus ellenőrzésével felfedezik az esetleges hibás részeket, s azokat hasonló típusú, készenléti állapotban található tartalék egységekkel cserélik ki.

zónázott decimális ábrázolás (unpacked decimal representation) – decimális számok ábrázolására használt olyan belső forma, amelynél minden →bájt egy decimális jegyet tartalmaz a jobb oldali →tetrádban, négy bináris 1-est (hexadecimális F) pedig a baloldaliban. Tkp. az →EBCDIC-kóddal megegyező formájú kódolás. Az előjel ábrázolására a jobb szélső bájt bal oldali tetrádját használják: a „+" számára a hexadecimális A, C, E és F, a „–" számára pedig a B és D vannak fenntartva. A ~t elsősorban a →b/k egységekben használják, a decimális aritmetikai műveletek elvégzésére a →tömörített decimális ábrázolás szolgál. Az →utasításkészletben mindig megtalálható az a speciális utasításpár, amely e két ábrázolási formátum közötti átalakításokat elvégzi.

ZS

zsebszámológép (calculator) - cigarettatárca nagyságú elektronikus számólógép, amely →integrált áramkörös felépítésének köszönhetően elemekkel is működtethető. Billentyűzettel rendelkezik az adatok és a műveletek kiválasztására, az eredmények pedig optikai kijelzőről közvetlenül leolvashatók. A legegyszerűbb ~ek a négy aritmetikai alapművelet, a százalékszámítás és a fordított érték elvégzésére alkalmasak, a fejlettebb típusok trigonometriai függvények és logaritmusok, négyzetgyökvonás stb. kiszámítását is lehetővé teszik. A →mikroprocesszorok megjelenésével olyan ~ek kifejlesztése is lehetővé vált, amelyekkel már bizonyos fokon programozni is lehet.

Angol – magyar szójegyzék

A

abnormal termination – rendellenes befejezés
abonent – előfizető
absolute address – abszolút cím
absolute binary form – abszolút bináris forma
absolute coding – abszolút kódolás
absolute error – abszolút hiba
absolute expression – abszolút kifejezés
absolute loader – abszolút betöltőprogram
abstract computer – absztrakt számítógép
abstract [data] type – absztrakt [adat]típus
abstract program – absztrakt program
abstract syntax – absztrakt szintaxis
access – elérés, hozzáférés
access key – hozzáférési kulcs
access level – hozzáférési szint
access method – hozzáférési módszer
access module – hozzáférési modul
access time – elérési idő
accumulator – akkumulátor
ACK (ACKnowledge) – visszaigazolás
acknowledge – visszaigazolás
acknowledge character – visszaigazolás karakter
acoustic coupler – akusztikus csatoló
acoustic memory – akusztikus memória
action entries – tevékenységjegyzék
action stub – tevékenységtörzs
active abonent – aktív előfizető
active computer – aktív számítógép
active file – aktív [adat]állomány
active page register – aktívlapregiszter
active program – aktív program
active star – aktív csillag
activity – aktivitás
actual key – aktuális kulcs, tényleges kulcs
actual parameter – aktuális paraméter
actual value – aktuális érték
adaptability – alkalmazkodóképesség
adaptive channel allocation – adaptív csatornakiosztás
adaptive routing – alkalmazkodó adatútképzés
ADC (Analogue-to-Digital Converter) – analóg-digitális konverter
additional record – kiegészítő rekord
address – cím
address bus – címsín
address computing – címszámítás
address constant – címkonstans
address expression – címkifejezés
address mapping – címátalakítás
address modification – címmódosítás
address part – címrész
address register – címregiszter
address representation – címábrázolás
address sorting – címrendezés
address space – címtartomány
address translation – címáthelyezés
addressability – címezhetőség
addressing – címzés
addressless instruction – cím nélküli utasítás
AI (Artificial Intelligence) – mesterséges intelligencia
algorithm – algoritmus
algorithmic function – algoritmikus függvény
algorithmic language – algoritmikus nyelv
alphabet – ábécé
alphanumerical – alfanumerikus
alphanumerical code – alfanumerikus kód
alphanumerical constant – alfanumerikus konstans
alphanumerical displaying – alfanumerikus kijelzés
alphanumerical terminal – alfanumerikus terminál
alternate routing – alternatív adatútképzés
alternative – alternatíva
ALU (Arithmetical and Logical Unit) – aritmetikai és logikai egység
ambigous rule – kétértelmű szabály
analogue computer – analóg számítógép
analogue computing – analóg számítás
analogue signal – analóg jel
analogue-to-digital converter – analóg-digitális konverter
analogue transmission – analóg adatátvitel
AND – ÉS
anisochronous signal – aszinkron jel
anonymous variable – anonim változó
application layer protocol – alkalmazói rétegprotokoll
application software – alkalmazói szoftver
a priorinei error – öröklött hiba
APSE (Ada Programming Support Environment) – programozási környezet
architecture – architektúra
argument – argumentum

arithmetical and logical unit – aritmetikai és logikai egység
arithmetical coprocessor – aritmetikai társprocesszor
arithmetical expression – aritmetikai kifejezés
arithmetical operation – aritmetikai művelet
arithmetical relation – aritmetikai reláció
arithmetical shift – aritmetikai léptetés
arithmetical unit – aritmetikai egység
ARQ (Automatic Repeat reQuest) – automatikus ismétléskérés
array – tömb
array [data] type – tömb [adat]típus
arrow key – nyílbillentyű
artificial intelligence – mesterséges intelligencia
artificial language – mesterséges nyelv
ASCII-code – ASCII-kód
ASCIIZ string – ASCIIZ-füzér
assembler – asszembler
assembler directive – asszembler direktíva
assembler listing – asszembler lista
assembly language – asszembler nyelv
assign – hozzárendelés
assignment statement – értékadó utasítás
associative addressing – asszociatív címzés
associative memory – asszociatív memória
associative register – asszociatív regiszter
assumed decimal point – feltételezett tizedespont
asynchronous – aszinkron
asynchronous channel – aszinkron csatorna
asynchronous communication – aszinkron adatátvitel
asynchronous computer – aszinkron számítógép
asynchronous modem – aszinkron modem
atom – atom
attribute – attribútum
attribute grammar – attribútumnyelvtan
AU (Arithmetical Unit) – aritmetikai egység
autocode – autokód
autocomplementing code – önkomplementáló kód
autoincremental addressing – növekményes címzés
automatically cleared failure – automatikusan javított hiba
automatic repeat request – automatikus ismétléskérés
automatic variable – automatikus változó
automatic volume recognition – automatikus kötetfelismerés
automaton – automata
auxiliary memory – segédmemória
availability – rendelkezésre állás
average access time – átlagos elérési idő
AVR (Automatic Volume Recognition) – automatikus kötetfelismerés
axiomatic semantics – axiomatikus szemantika

B

background colour – háttérszín
background partition – háttérpartíció
background processing – háttérfeldolgozás
back reading – visszafelé olvasás
backspace – visszalépés
backtracking – visszalépegetés
back-up copy – visszaállítási másolat
Backus-Naur Form – Backus-Naur Forma
balanced merging – egyenletes összeválogatás
balanced tree – kiegyensúlyozott fa
BAM (Basic Access Method) – alap-adatelérési technika
bandwith – sávszélesség
bar code – vonalkód
base address – báziscím
baseband networking – alapsávú adatátvitel
baseband signal – alapsávi jel
base language – bázisnyelv
base machine – alapgép
base register – bázisregiszter
basic access method – alap-adatelérési technika
basic direct access method – direkt alap-adatelérési technika
basic indexed-sequential access method – indexszekvenciális alap-adatelérési technika
basic partitioned access method – particionált alap-adatelérési technika
basic sequential access method – szekvenciális alap-adatelérési technika
basic symbol – alapszimbólum

basic telecommunication access method – alap-távfeldolgozási mód
batch processing – adagolt feldolgozás, kötegelt feldolgozás
batch processing terminal – kötegelt feldolgozású terminál
baud – baud
BCC (Block Check Character) – blokkellenőrző karakter
BCD (Binary Coded Decimal) – binárisan kódolt decimális kód
BDAM (Basic Direct Access Method) – direkt alap-adatelérési technika
benchmark – berendezéskiértékelés
biased exponent – eltolt kitevő
binary addition – bináris összeadás
binary code – bináris kód
binary coded decimal code – binárisan kódolt decimális kód
binary complement – bináris komplementum
binary counter – bináris számláló
binary loader – bináris betöltőprogram
binary multiplication – bináris szorzás
binary number – bináris szám
binary pattern – bináris mintázat
binary point – bináris pont
binary representation – bináris ábrázolás
binary search – bináris keresés
binary search tree – bináris keresési fa
binary tree – bináris fa
binary-to-decimal conversion – bináris-decimális konverzió
binary-to-hexadecimal conversion – bináris-hexadecimális konverzió
binary-to-octal conversion – bináris-oktális konverzió
bipolar coding – bipoláris kódolás
BISAM (Basic Indexed Access Method) – indexszekvenciális alap-adatelérési technika
bit – bit
bit density – bitsűrűség
bit mapping – bittérképezés
bit-parallel transmission – bitpárhuzamos átvitel
bit pattern – bitminta
bit-serial transmission – bitsoros átvitel
bit slice – bitszelet
bit-slice architecture – bitszelet-architektúra
bits per inch – bpi
bits per second – bps
bit stuffing – bitbeültetés
blinking – villogás
block – blokk
block check character – blokkellenőrző karakter
block diagram – blokkdiagram
block ignore character – blokk-kizáró karakter
blocking – blokkolás
blocking factor – blokkolási tényező
block structure – blokkstruktúra
BNF (Backus-Naur Form) – Backus-Naur Forma
Boolean algebra – Boole-algebra
Boolean [data] type – Boole-[adat]típus
Boolean function – Boole-függvény
boot sector – betöltőszektor
bootstrap loader – elsődleges betöltőprogram
borrow – kölcsönvétel
bottom-up method – alulról felfelé haladó módszer
bottom-up syntactical analysis – alulról felfelé haladó szintaktikai elemzés
BPAM (Basic Partitioned Access Method) – particionált alap-adatelérési technika
branch – elágazás
branch instruction/statement – ugró utasítás
breakpoint – felfüggesztési pont
BSAM (Basic Sequential Access Method) – szekvenciális alap-adatelérési technika
BTAM (Basic Telecommunication Method) – alap-távfeldolgozási mód
bubble sort – buborékrendezés
buffer – puffer
buffering – puffertechnika
buffer memory – puffermemória
buffer register – pufferregiszter
built-in macroinstruction – beépített makroutasítás
burning – beégetés
bus – busz, sínrendszer
byte – bájt
byte multiplexing – bájtmultiplexelés
byte stuffing – bájtbeültetés
byte-organized computer – bájtszervezésű számítógép
byte-parallel transmission – bájtparallel átvitel
byte-serial transmission – bájtsoros átvitel
bus, highway – sínrendszer

C

cache memory – gyorsítótár, előtároló
CAD (Computer-Aided Design) – számítógéppel segített tervezés
CAI (Computer-Aided Instruction) – számítógéppel segített oktatás
calculator – számológép, zsebszámológép
call – hívás
call by address – cím szerinti hivatkozás
call by name – név szerinti hivatkozás
call by value – érték szerinti hivatkozás
calling sequence – hívási sor
CAM(Computer-Aided Manufacturing) – számítógéppel segített gyártás
CAM (Content-Addressed Memory) – asszociatív memória
CAMAC (Computer Aided Measurement And Control) – számítógéppel segített mérés és vezérlés
card puncher/perforator – kártyalyukasztó
card reader – kártyaolvasó
card-to-tape converter – kártya-szalag konverter
carriage – kocsi
carriage-return character – „kocsi vissza" karakter
carry – átvitel
cartridge disk – kazettás mágneslemez
cataloged procedure – katalogizált eljárás
catalogue – katalógus
cathod ray tube displaying device – katódsugárcsöves megjelenítő
CB (Command Block) – vezérlőblokk
CCW (Channel Command Word) – csatornaparancs
CD-ROM (Compact Disk Read Only Memory) – kompaktlemez
centralized processing – centralizált feldolgozás
central memory – központi memória
central monitor – központi monitor
central processing unit – központi vezérlőegység
central terminal – központi terminál
central unit – központi egység
chaining – láncolás
change file – változás-[adat]állomány
channel – csatorna
channel command – csatornaparancs
channel program – csatornaprogram
channel status word – csatorna-állapotszó
character – karakter
character [data] type – karakter [adat]típus
characteristic – karakterisztika
characteristic overflow – karakterisztika-túlcsordulás
characteristic underflow – karakterisztika-alulcsordulás
character recognition – karakterfelismerés
character set – karakterkészlet
character string – karakterlánc
checkpoint – ellenőrzőpont
checksum – kontrollösszeg
chip – morzsa
Chomsky-hierarchy – Chomsky-hierarchia
circular list – körkörös lista
circular shift – körkörös léptetés
CISC (Complex Instruction Set Computer) – komplex utasításkészletű számítógép
class condition – típusfeltétel
class test – típusteszt
clause – klauzula
closed-loop control – zárt hatásláncú vezérlés
closed shop – zárt gépterem
closed subroutine – zárt szubrutin
cluster – fürt
CNC (Computerized Numeric Control) – számítógépes numerikus vezérlés
CNF (Conjuctive Normal Form) – konjuktív normálalak
code – kód
code converter – kódkonverter
code generation – kódgenerálás
code optimization – kódoptimalizálás
cold start – hidegindítás
combined condition – összetett feltétel
command block – vezérlőblokk
command chaining – parancsláncolás
command file – parancsállomány
command interpreter – parancsértelmező
command language – parancsnyelv
command tape – vezérszalag
comment – magyarázó szöveg
common segment – közös szegmens
common sense knowledge – hétköznapi tudás
communication – kommunikáció
communication area – kommunikációs terület

communication channel – kommunikációs csatorna
communication interface – kommunikációs interfész
communication processor – kommunikációs processzor
communication software – adatátviteli szoftver
compact disk – kompaktlemez
comparison – összehasonlítás
compatibility – kompatibilitás
compilation unit – fordítási egység
compiler – fordítóprogram, kompilátor
compiler writing language – fordítóprogram-író nyelv
compile time error – fordítási hiba
compiling – fordítás
compiling time – fordítási idő
complement code – komplemens kód
complex [data] type – komplex [adat]típus
complex instruction set computer – komplex utasításkészletű számítógép
computed address – számított cím
computer – számítógép
computer-aided design – számítógéppel segített tervezés
computer-aided instruction – számítógéppel segített oktatás
computer-aided manufacturing – számítógéppel segített gyártás
computer aided measurement and control – számítógéppel segített mérés és vezérlés
computer generation – számítógép-generáció
computerized numeric control – számítógépes numerikus vezérlés
computer graphics – számítógépes grafika
computer network – számítógép-hálózat
computer series – számítógépcsalád
computing technique – számítástechnika
concatenation – konkatenáció, computerized numeric control – számítógépes numerikus vezérlésösszeláncolás
concrete syntax – konkrét szintaxis
concurrent Pascal – konkurens Pascal
concurrent system – konkurens elvű rendszer
condition – feltétel
conditional branch – feltételes elágazás
conditional compiling – feltételes fordítás
conditional control instruction – feltételes vezérlésátadó utasítás
conditional statement – feltételes utasítás
conditional variable – feltételes változó
condition entries – feltételjegyzék
condition name – feltételnév
condition name condition – feltételnév-feltétel
condition stub – feltételtörzs
configuration – konfiguráció
configuration file – konfigurációs állomány
congestion – torlódás
conjunction – konjunkció
conjunctive normal form – konjunktív normálalak
console typewriter – konzolírógép
constant – konstans
contention – versenyhelyzet
context-free grammar – környezetfüggetlen nyelvtan
context-free language – környezetfüggetlen nyelv
context-sensitive grammar – környezetfüggő nyelvtan
context-sensitive language – környezetfüggő nyelv
control bit – ellenőrzőbit
control card – vezérkártya
control character – vezérlőkarakter
control digit – ellenőrző számjegy
control information – vezérlési információ
controlled execution – ellenőrzött végrehajtás
controller – vezérlő
control module – vezérlőmodul
control statement – vezérlésátadó utasítás
control structure – vezérlőszerkezet
control unit – vezérlőegység
convergence – konvergencia
conversational compiling – párbeszédes fordítás
conversational mode – párbeszédes üzemmód
conversational terminal – párbeszédes terminál
conversion error – konverziós hiba
coprocessor – társprocesszor
core memory – ferritmemória
coroutines – korutinok

CPU (Central Processing Unit) – központi vezérlőegység
CR (Carriage Return) – „kocsi vissza" karakter
CRAM (Card Random Access Memory) – mágneskártya
CRC (Cyclic Redundancy Check) – ciklikus redundancia-ellenőrzés
created identifier – előállított azonosító
critical resource – kritikus erőforrás
critical section – kritikus szakasz
cross-assembler – keresztasszembler
cross-compiler – keresztfordító
cross reference list – kereszthivatkozások listája
cross-software – keresztszoftver
CRT (Cathode Ray Tube terminal) – katódsugárcsöves megjelenítő
CSW (Channel Status Word) – csatorna-állapotszó
CU (Control Unit) – központi egység
cursor – kurzor, helyőr
cybernetics – kibernetika
cycle – ciklus
cycle stealing – cikluslopás
cyclic code – ciklikus kód
cyclic redundancy check – ciklikus redundancia-ellenőrzés
cyclic shift – ciklikus léptetés
cylinder – cilinder, palást
cylinder index – cilinderindex

D

data – adat
data abstraction – adatabsztrakció
data access method – adatelérési technika
data acquisition – adatgyűjtés
data acquisition system – adatgyűjtő rendszer
data bank – adatbank
data base – adatbázis
data-base administrator – adatbázis-felügyelő
data-base copy – adatbázis-másolat
data-base language – adatbázisnyelv
data-base management system – adatbázis-kezelő rendszer
data-base network – adatbázis-hálózat
data-base recovery – adatbázis-visszaállítás
data-base schema – adatbázisséma
data-base subschema – adatbázis-alséma
data-base tuning – adatbázis-hangolás
data block – adatblokk
data bus – adatsín
data chaining – adatláncolás
data communication network – adatátviteli hálózat
data concentration method – adattömörítési módszer
data concentrator – adatkoncentrátor
data consistency – adatkonzisztencia
data contamination – adatfertőzés
data control block – adatvezérlő blokk
data conversion – adatkonverzió
data description language – adatleíró nyelv
data dictionary – adatszótár
data division – adatleíró főrész
data element – adatelem
data hierarchy – adathierarchia
data independence – adatfüggetlenség
data initialization statement – kezdőértékadó utasítás
data integrity – adatintegritás
data link – adatkapcsolat
data link protocol – adatkapcsolati rétegprotokoll
data manipulation language – adatkezelő nyelv
data media – adathordozó
data migration – adatvándorlás
data model – adatmodell
data module – adatmodul
data name – adatnév
data network – adathálózat
data packet – adatcsomag
data packet interleaving – adatcsomag-egymásbaágyazás
data preparation – adat-előkészítés
data privacy – adattitkosság
data processing – adatfeldolgozás
data processing machine – adatfeldolgozó gép
data security – adatbiztonság
data segment – adatszegmens
data structure – adatstruktúra
data transfer operation – adatátviteli művelet
data transmission rate – adatátviteli sebesség
[data] type – [adat]típus
data validation – adatérvényesítés

DB (Data Base) – adatbázis
DBMS (Data Base Management System) – adatbázis-kezelő rendszer
DDB (Distributed Data Base) – elosztott adatbázis
DDL (Data Description Language) – adatleíró nyelv
DDP (Distributed Data Processing) – elosztott feldolgozás
deadlock – holtpont,végleges kizárás
debugging program – hibafelderítő program
decimal representation – decimális számábrázolás
decimal-to-binary conversion – decimális-bináris konverzió
decimal-to-hexadecimal conversion – decimális-hexadecimális konverzió
decimal-to-octal conversion – decimális-oktális konverzió
decision rule – döntési szabály
decision table – döntési táblázat
decision tree – döntési fa
declaration – deklaráció
declarative language – deklaratív nyelv
declarative statement – deklaratív utasítás
decoder – dekódoló
decoding – dekódolás
dedicated resource – dedikált erőforrás
default value – alapértelmezési érték
definition – definíció
delay vector – késleltetési vektor
delimiter – elhatárolójel
dependent condition system – függő feltételrendszer
derivation rule – helyettesítési szabály
derivative syntactical analysis – levezető szintaktikai elemzés
derived data item – származtatott adatelem
derived [data] type – származtatott [adat]típus
descriptive model – leíró modell
desk-top computer – asztali számológép
desktop publishing – kiadványszerkesztő
destructive memory – destruktív memória
destructive reading – törlő olvasás
deterministic automaton – determinisztikus automata
deterministic game – determinisztikus játék
deterministic program – determinisztikus program
device controller – eszközvezérlő
diagnostic program – diagnosztizáló program
digit – számjegy
digital computer – digitális számítógép
digital printer – digitális nyomtató
digital signal – digitális jel
digital-to-analogue converter – digitális-analóg konverter
digital transmission – digitális adatátvitel
digitization – digitalizálás
direct access file – közvetlen hozzáférésű [adat]állomány
direct access method – közvetlen hozzáférési módszer
direct code – direkt kód
direct file – direkt szervezésű [adat]állomány
direct-insert subroutinr – nyílt szubrutin
directive – direktíva
direct memory access – közvetlen tárhozzáférés
direct memory access unit – közvetlen tárhozzáférési egység
directory routing – címlista szerinti adatútképzés
discrete variable – diszkrét változó
disjunction – diszjunkció
disjunctive normal form – diszjunktív normálalak
disk operating system – lemezes operációs rendszer
displacement – eltolás
displaying – kijelzés
displaying device – kijelző, megjelenítő
distributed control – elosztott vezérlés
distributed data base – elosztott adatbázis
distributed processing – elosztott feldolgozás
distribution sorting – szétszórásos rendezés
DMA (Direct Memory Access) – közvetlen tárhozzáférés
DMA-controller – közvetlen tárhozzáférési egység
DML (Data Manipulation Language) – adatkezelő nyelv
DNF (Disjunctive Normal Form) – diszjunktív normálalak
document – bizonylat
document reader – bizonylatolvasó
domain expert – tárgyköri szakértő
double precision – dupla pontosság
doubly linked list – kétszeresen láncolt lista

DP (Data Processing) – adatfeldolgozás
driver – meghajtó
DTP (DeskTop Publishing) – kiadványszerkesztő
dual system – duálrendszer
dummy address – álcím
dummy instruction – álutasítás
dummy section – álszakasz
duplex channel – duplex csatorna
duplex communication – duplex összeköttetés
dynamic attribute – dinamikus attribútum
dynamic data structure – dinamikus adatstruktúra
dynamic memory allocation – dinamikus memóriakiosztás
dynamic model – dinamikus modell
dynamic multiplexing – dinamikus multiplexelés
dynamic relocation – dinamikus áthelyezés
dynamic subroutine – dinamikus szubrutin
dynamic variable – dinamikus változó

E

EBCDIC-code – EBCDIC-kód
echo – visszhang
echo checking – visszhangellenőrzés
editing – szerkesztés
editing character – szerkesztési karakter
EEROM (Electrically Erasable Read-Only Memory) – elektromosan törölhető permanens memória
effective address – tényleges cím, effektív cím
electrically erasable read-only memory – elektromosan törölhető permanens memória
electronic mail – elektronikus posta
electrostatic memory – elektrosztatikus memória
electrostatic printer – elektrosztatikus nyomtató
„else" roule – „egyébként"szabály
embedded code – beágyazott kód
embedded system – beágyazott rendszer
empirical formula – empirikus formula
EMS (Electronic Mail System) – elektronikus posta
emulation – emulálás
encoding – kódolás
endless loop – végtelen ciklus
end-of-block character – blokkvége karakter
end-of-file label – [adat]állomány-zárócímke
end-of job – munkavége
end-of message character – üzenetvége karakter
end-of-text character – szövegvége karakter
end-of-transmission character – adásvége karakter
end-of volume label – kötet-zárócímke
end-user – végső felhasználó
ENQ (ENQuiry) – kérdezés karakter
enquiry character – kérdezés karakter
entity – entitás
entry point – belépési pont
enumerated [data] type – felsorolt [adat]típus
EOB (End-Of Block) – blokkvége karakter
EOF (End-Of File) – [adat]állomány-zárócímke
EOJ (End-Of Job) – munkavége
EOM (End-Of Message) – üzenetvége karakter
EOT (End-Of Transmission) – adásvége karakter
EOV (End-Of Volume) – kötet-zárócímke
EPROM (Erasable Programmable Read-Only Memory) – törölhető-programozható permanens memória
equivalence – ekvivalencia
erasable programmable read-only memory – törölhető-programozható permanens memória
error – hiba
error analysis – hibaelemzés
error burst – hibacsomó
error check bit – hibajelző bit
error correcting code – hibajavító kód
error detecting code – hibajelző kód
error estimation – hibabecslés
error formula – hibaképlet
error indication – hibajelzés
error level – hibaszint
error limiting – hibakorlátozás
error limiting encoding – hibakorlátozó kódolás
error limiting strategy – hibakorlátozó stratégia
error list – hibalista
error message – hibaüzenet

error of formula – képlethiba
error recovery – hiba utáni visszaállítás
estimation – becslés
ETX (End-of TeXt) – szövegvége karakter
exception – kivétel
exclusive condition system – kizáró feltételrendszer
exclusive OR – kizáró VAGY
executable form – közvetlenül végrehajtható forma
executable statement – végrehajtható utasítás
execution error – végrehajtási hiba
execution time – végrehajtási idő
existential quantifier – egzisztenciális kvantor
expert system – szakértőrendszer
explanatory model – magyarázó modell
extended machine – kiterjesztett gép
extensible language – kiterjeszthető nyelv
extension card – bővítőkártya
external data – külső adat
external interruption – külső megszakítás
external memory – külső memória
external reference – külső hivatkozás
external sorting – külső rendezés

F

facsimile – fakszimile
failure journal – hibanapló
FAT (File Allocation Table) – [adat]állomány-kiosztási táblázat
fatal error – végzetes hiba
FDC (Floppy Disk Controller) – hajlékonylemezes vezérlő
FDD (Floppy Disk Drive) – hajlékonylemezes meghajtó
FDM (Frequency Division Multiplexing) – frekvenciaosztásos multiplexelés
ferrite core – ferritgyűrű
field – mező
FIFO-structure – FIFO-szerkezet
figurative constant – figuratív konstans
file – [adat]állomány
file activity ratio – [adat]állomány-aktivitási hányados
file allocation table – [adat]állomány-kiosztási táblázat
file closing – [adat]állomány-lezárás
file creating – [adat]állomány-létrehozás
file [data] type – állomány [adat]típus
file header label – [adat]állomány-kezdőcímke
file identification – [adat]állomány-azonosítás
file inventory – [adat]állomány-kivonatolás
file maintenance – [adat]állomány-karbantartás
file management – [adat]állomány-kezelés
file management system – [adat]állomány-kezelő rendszer
file name – [adat]állománynév
file name extension – [adat]állománynév-kiterjesztés
file opening – [adat]állomány-megnyitás
file searching – [adat]állomány-keresés
file updating – [adat]állomány-aktualizálás
filter – szűrő
finite automaton – véges automata
finite-element method – végeselemek módszere
firmware – firmver
first approximation – első közelítés
first generation computer – első generációs számítógép
fixed length record – rögzített hosszúságú rekord
fixed memory – permanens memória
fixed-partitions method – rögzített partíciók módszere
fixed-point operation – fixpontos művelet
fixed-point representation – fixpontos számábrázolás
flag – jelző
flexibility – rugalmasság
floating-point arithmetic unit – lebegőpontos aritmetikai egység
floating-point coeficient – mantissza
floating-point operation – lebegőpontos művelet
floating-point representation – lebegőpontos számábrázolás
floating-point subrutine – lebegőpontos szubrutin
flood routing – elárasztásos adatútképzés
floppy disk – hajlékonylemez
flowchart – folyamatábra
flow control – adatáramlás-vezérlés
font – betűkészlet
foreground colour – előtérszín
foreground partition – előtér-partíció
formal language – formális nyelv
formal parameter – formális paraméter

formal system – formális rendszer
formatting – formázás
formula manipulation language – formulakezelő nyelv
forward error correction – közvetlen hibajavítás
four-address computer – négycímű számítógép
fourth-generation computer – negyedik generációs számítógép
frame – keret
frame-based method – keret alapú módszer
„from left to right" rule – „balról jobbra" szabály
front-end computer – „front-end" számítógép
full duplex channel – teljes duplex csatorna
function – függvény
functional dependency – funkcionális függőség
functional key – funkcionális billentyű

G

game – játék
game of hazard – szerencsejáték
games theory – játékelmélet
gap – blokk-köz
garbage collection – hulladékgyűjtés
generalized information system – általánosított információs rendszer
general-purpose computer – általános célú számítógép
general register – általános regiszter
generation number – generációszám
generative grammar – generatív nyelvtan
generic unit – generikus egység
global error – globális hiba
global symbol – globális szimbólum
global symbol table – globális szimbólumtáblázat
goal symbol – kezdőszimbólum
grammar – grammatika, nyelvtan
grammatical constant – nyelvtani konstans
grammatical variable – nyelvtani változó
graph – gráf
graphic data processing – grafikus adatfeldolgozás
graphic display – grafikus kijelző
graphic displaying – grafikus kijelzés
graphic mode – grafikus mód
graphic pattern generation – grafikus formák generálása
graphic pattern recognition – grafikus alakfelismerés
graphic terminal – grafikus terminál

H

half duplex channel – félduplex csatorna
half duplex communication – félduplex összeköttetés
handshaking – kölcsönös szinkronizálás, kézfogás
hard copy – dokumentummásoló
hard disk – merevlemez
hardware – hardver
hashing – terítőfüggvény
HDC (Hard Disk Controller) – merevlemezes vezérlő
HDD (Hard Disk Drive) – merevlemezes meghajtó
HDR (HeaDeR label) – [adat]állomány-kezdőcímke
header – fejléc
help file – segélyállomány
heterogeneous data structure – heterogén adatstruktúra
heuristic program – heurisztikus program
heuristic programming – heurisztikus programozás
heuristic routing – heurisztikus adatútképzés
heuristics – heurisztika
hexadecimal representation – hexadecimális ábrázolás
hidden-line algorithm – takartvonal-algoritmus
hierarchical structured data – hierarchikusan strukturált adatok
high level programming language – magas szintű programozási nyelv
HLL (High Level Language) – magas szintű programozási nyelv
home computer – házi számítógép
homogeneous data structure – homogén adatstruktúra
host computer – „honos" számítógép, gazdaszámítógép
host language – befogadó nyelv

„hot potato" routing – „forró krumpli" adatútképzés
housekeeping – rezsiművelet
HSI (High Scale Integration) – magas fokú integráltság
Huffmann-code – Huffmann-kód
hybrid computer – hibrid számítógép

I

IC (Integrated Circuit) – integrált áramkör
icon – ikon
identification division – azonosító főrész
identifier – azonosító
illegal character – illegális karakter
illegal instruction – illegális utasítás
image processing – képfeldolgozás
image refreshing – képfrissítés
immediate operand – közvetlen operandus
implementation dependency – implementációfüggőség
implication – implikáció
implied addressing – implicit címzés
incremental compiling – inkrementális fordítás
incrementation – inkrementálás
independent condition system – független feltételrendszer
index – index^2
indexed address – indexelt cím
indexed-sequential access method – indexszekvenciális hozzáférési módszer
indexed-sequential file – indexszekvenciális [adat]állomány
index file – indexállomány
index register – indexregiszter
indirect access – közvetett hozzáférés
indirect address – indirekt cím, közvetett cím
inference engine – következtető gép
infinite loop – végtelen ciklus
infix form – infix forma
informatics – informatika
information – információ
information hiding – információrejtés
information quality – információminőség
information rate – információhozam
information representation – információábrázolás
information retrieval – információ-visszakeresés
information's amount – információmennyiség
information saturation – információtelítettség
information theory – információelmélet
information threshold – információküszöb
inherited attribute – öröklött attribútum
inherited error – örökölt hiba
initialization – inicializálás
initiating routine – nyitószabály
input alphabet – bemenő ábécé
input file – bemeneti [adat]állomány
input-output channel – adatcsatorna
input unit – bemeneti egység
inquiry – lekérdezés
insertion sorting – beszúrásos rendezés
instruction – utasítás
instruction activation – utasításaktivizálás
instruction code – utasításkód
instruction counter – utasításszámláló regiszter
instruction cycle – utasításciklus
instruction decoder – utasításdekódoló
instruction execution – utasítás-végrehajtás
instruction fetch – utasításlehívás
instruction interpretation – utasításértelmezés
instruction register – utasításregiszter
instruction set – utasításkészlet
iinteger constant – egész konstans
nteger [data] type – egész [adat]típusi
ntegrated circuit – integrált áramkör
integrated information system – integrált információs rendszer
integrated services digital network – integrált szolgáltatású digitális hálózat
integration – integráció
intelligent terminal – intelligens terminál
interactive graphics – interaktív grafika
interactive processing – interaktív feldolgozás
interactive terminal – interaktív terminál
interchange sorting – helycserés rendezés
interface – interfész, illesztőegység
interlock – kölcsönös kizárás
intermediate language – közbülső nyelv
internal code – belső kód
internal data – belső adat
internal memory – belső memória
internal sorting – belső rendezés
interpretation – értelmezés

interpreter – értelmezőprogram
interrupt – megszakítás
interruptable instruction – megszakítható utasítás
interruptable state – megszakítható állapot
interrupt address – megszakítási cím
interrupted program – megszakított program
interrupt level – megszakítási szint
interrupt masking – megszakításmaszkolás
interrupt request – megszakításkérés
interrupt service subroutine – megszakításfeldolgozó szubrutin
interrupt system – megszakítási rendszer
inverted file – invertált [adat]állomány
i/o bus – b/k gyűjtősín
i/o module – b/k modul
i/o operation – b/k művelet
i/o processor – b/k processzor
i/o unit – b/k egység
IRQ (Interrupt ReQuest) – megszakításkérés
ISAM (Indexed-Sequential Access Method) – indexszekvenciális hozzáférési módszer
ISDN (Integrated Services Digital Network) – integrált szolgáltatású digitális hálózat
iteration statement – ciklusutasítás
iterative method – iterációs eljárás

J

Jackson's method – Jackson-módszer
JCL (Job Control Language) – munkavezérlő nyelv
job – munka
job control language – munkavezérlő nyelv
job file – munka-[adat]állomány
job scheduler – munkaütemező
joystick – botkormány

K

key – kulcs
keyboard – billentyűzet
keyword – kulcsszó
knowledge – ismeret
knowledge base – ismeretbázis
knowledge engineer – ismerettechnológus
knowledge engineering – ismerettechnológia
knowledge representation – ismeretábrázolás

L

label – címke
label constant – címkekonstans
label variable – címkeváltozó
LAN (Local Area Network) – helyi hálózat
language – nyelv
language processor – nyelvi processzor
laptop computer – hordozható számítógép
large scale of integration – nagyfokú integráltság
lazy evaluation – visszafogott kiértékelés
LCB (Line Control Block) – vonalvezérlő blokk
leased line – bérelt vonal
left justified representation – balra igazodó ábrázolás
letters-figures shift – betű-szám váltás
lexical analysis – lexikális analízis
lexical analyzer, scanner – lexikális elemző
LF (Line Feed) – újsor karakter
librarian – könyvtárkezelő program
library – könyvtár
LIFO-structure – LIFO-szerkezet
light pen – fényceruza
limited entry – korlátozott bejegyzés
limit of error – hibakorlát
limit register – határregiszter
linear data structure – lineáris adatstruktúra
linear grammar – lineáris nyelvtan
linear programming – lineáris programozás
linear search – lineáris keresés
line concentration – vonalkoncentrálás
line control block – vonalvezérlő blokk
line feed character – újsor karakter
line multiplexing – vonalmultiplexelés
line switching – vonalkapcsolás
line switching network – vonalkapcsolásos hálózat
link – összeköttetés
linkage editor – kapcsolóprogram
linked file – láncolt [adat]állomány
linking – kapcsolás
list – lista
listing – listázás
list processing – listafeldolgozás
literal – literál

loading – betöltés
loading address – betöltési cím
loading program – betöltőprogram
loading time – betöltési idő
local area network – helyi hálózat
local error – helyi hiba
local symbol – lokális szimbólum
locate mode – helymeghatározó mód
logarithmic search – logaritmikus keresés
logical access – logikai hozzáférés
logical addition – logikai összeadás
logical address – logikai cím
logical address space – logikai címtartomány
logical algebra – logikai algebra
logic-based method – logikai alapú módszer
logical constant – logikai konstans
logical data structure – logikai adatszerkezet
logical [data] type – logikai [adat]típus
logical device name – logikai eszköznév
logical error – logikai hiba
logical expression – logikai kifejezés
logical file – logikai [adat]állomány
logical function – logikai függvény
logical multiplication – logikai szorzás
logical-physical description – logikai-fizikai leírás
logical programming – logikai programozás
logical record – logikai rekord
logical segment – logikai szegmens
logical shift – logikai léptetés
logical unit – logikai egység
logical value – logikai érték
logical variable – logikai változó
log in – bejelentkezés
log off – kijelentkezés
low-level language – alacsony szintű nyelv
LSI (Large Scale of Integration) – nagyfokú integráltság

M

machine code – gépi kód
machine cycle – gépi ciklus
machine independent – gépfüggetlen
machine instruction – gépi utasítás
machine language – gépi nyelv
machine processible form – géppel feldolgozható forma
machine translation – gépi fordítás
macroassembler – makroasszembler
macrocall – makrohívás
macrodefinition – makrodefiníció
macroexpansion – makrokifejtés
macroinstruction – makroutasítás
macrolanguage – makronyelv
macrolibrary – makrokönyvtár
macroparameter – makroparaméter
macroprocessing – makrofeldolgozás
macroprocessor – makroprocesszor
macrospecification – makromeghatározás
magnetic card – mágneskártya
magnetic disk – mágneslemez
magnetic disk storage – mágneslemezes memória
magnetic tape – mágnesszalag
magnetic tape organization – mágnesszalagos szervezés
main memory – elsődleges memória
main program – főprogram
maintenance program – karbantartó program
malfunction program – zavarérzékelő program
MAN (Metropolitan Area Network) – városi hálózat
mantissa – mantissza, floating point coefficient
mask – maszk
masked interruption – maszkolt megszakítás
masking – maszkolás
mass memory – tömegtár
master file – törzs[adat]állomány
master index – főindex
mathematical logic – matematikai logika
MDA (Monochrome Display Adapter) – monokróm képernyőadapter
mean time between failures – hibák közötti átlagos idő
memory – memória
memory allocation – memóriakiosztás
memory capacity – memóriakapacitás
memory cycle – memóriaciklus
memory dump – memóriakiíratás
memory hierarchy – memóriahierarchia
memory image format – bináris tárkép forma
memory location – memórialokáció
memory management – memóriagazdálkodás
memory map – memóriatérkép
memory printout – memóriakiírás

memory protection – memóriavédelem
memory size – memóriakapacitás
memory word – memóriaszó
menu – menü
menu-driven mode – menüvezérelt üzemmód
merge sorting – összeválogatásos rendezés
merging – összeválogatás
message – üzenet
message concentrator – üzenetkoncentrátor
message editing – üzenetszerkesztés
message packet – üzenetcsomag
message queueing – üzenet-sorbaállítás
message switching – üzenetkapcsolás
message switching network – üzenetkapcsoló hálózat
metacharacter – metakarakter
metaconstant – metakonstans
metaexpression – metakifejezés
metalanguage – metanyelv
metasymbol – metaszimbólum
metavariable – metaváltozó
method error – módszerhiba
metropolitan area network – városi hálózat
microcomputer – mikroszámítógép
microinstruction – mikroutasítás
microoperation – mikroművelet
microprocessor – mikroprocesszor
microprogram – mikroprogram
microprogrammed computer – mikroprogramozott számítógép
microprogrammed control – mikroprogramozott vezérlés
microprogrammed logic – mikroprogramozott logika
minimax principle – minimax-tétel
minimum weight routing – minimálsúlyozású adatútképzés
mix – mix
mixed expression – vegyes kifejezés
mnemonic code – mnemonikus kód
model – modell
modem – modem
modified complement code – módosított komplemens kód
modular programming – moduláris programozás
module – modul
module library – modulkönyvtár
modulo-N addition – modulo-N összeadás
monitor – monitor
monochrome display adapter – monokróm képernyőadapter
monoprogramming – monoprogramozás
Monte-Carlo method – Monte-Carlo módszer
mother board – alaplap
mouse – egér
move mode – mozgató mód
MTBF (Mean Time Between Failures) – hibák közötti átlagos idő
multiaccess – többszörös hozzáférés
multiplexer – multiplexer
multiplexer channel – multiplexcsatorna
multiplexing – multiplexelés
multipoint network – felfűzéses hálózat
multiprocessing – multiprocesszálás
multiprocessor – multiprocesszor
multiprocessor system – többprocesszoros rendszer
multiprogramming – multiprogramozás

N

NAK (Negative AcKnowledge) – negatív visszaigazolás
name space – névtartomány
navigation – navigálás
NC (Numerical Control) – numerikus vezérlés
negation – negáció
negative acknowledge – negatív visszaigazolás
nesting – egymásbaágyazás
network control – hálózatvezérlés
network layer protocol – hálózati rétegprotokoll
next state equation – állapotegyenlet
nibble – tetrád
node – csomópont
non-deterministic automaton – nemdeterminisztikus automata
non-deterministic program – nemdeterminisztikus program
non-executable statement – nemvégrehajtható utasítás
non-linear programming – nemlineáris programozás
non-numerical literal – nemnumerikus literál
non-procedural language – nemprocedurális nyelv

non-standard label – nemszabványos címke
non-terminal symbol – nemterminális szimbólum
normalization – normalizálás
numerical analysis – numerikus analízis
numerical code – numerikus kód
numerical computer – numerikus számítógép
numerical constant – numerikus konstans
numerical control – numerikus vezérlés
numerical control of tools – szerszámgépek numerikus vezérlése
numerical data – numerikus adatok
numerical literal – numerikus literál
numerical method – numerikus módszer
numerical stability – numerikus stabilitás

O

object-based method – objektum alapú módszer
object language – tárgynyelv
object module – tárgymodul
object-oriented language – objektumorientált nyelv
object program – tárgyprogram
object program assembling – tárgyprogramszerkesztés
OCR (Optical Character Reader) – optikai karakterolvasó
OCR (Optical Character Recognition) – optikai karakterfelismerés
octal representation – oktális számábrázolás, nyolcas számábrázolás
octet – oktett
office automation – irodagépesítés
one-address computer – egycímű számítógép
one-level memory – egyszintű memória
one-way list – egyirányú lista
open-loop control – nyílt hatásláncú vezérlés
open shop – nyitott gépterem
open subroutine – nyílt szubrutin
open system interconnection – nyílt rendszerű összekapcsolás
operand – operandus
operand fetch – operanduslehívás
operating system – operációs rendszer
operation – művelet
operations research – operációkutatás
operative address space – operatív címtartomány
operative memory – operatív memória
operator – operátor
operator's console/keyboard – vezérlőpult
operator's intervention – gépkezelői beavatkozás
optical character reader – optikai karakterolvasó
optical character recognition – optikai karakterfelismerés
optical disk – optikai lemez
optimization – optimalizálás
optional word – opcionális szó
OR – VAGY
order key – besorolási kulcs
ordinal [data] type – sorszámozott [adat]típus
ordinal function – sorszámfüggvény
oriented graph – irányított gráf
orthogonal data structure – ortogonális adatstruktúra
OS (Operating System) – operációs rendszer
oscillate merging – oszcilláló összeválogatás
OSI (Open System Interconnection) – nyílt rendszerű összekapcsolás
output – kivitel
output alphabet – kimenő ábécé
output equation – kimeneti függvény
output file – kimeneti [adat]állomány
output unit – kimeneti egység
overflow – túlcsordulás
overflow bit – túlcsordulásbit
overhead – többletráfordítás
overlay – átfedés
overlay structure – átfedési szerkezet

P

package – csomag
packed decimal representation – tömörített decimális ábrázolás
packet switching – csomagkapcsolás
pad character – töltőkarakter
page – lap
paged addressing – lapozott címzés
page fault – laphiba
page fixing – laprögzítés

page replacement algorithm – laphelyettesítési algoritmus
page table – laptábla
paging – lapkezelés
parallel algorithm – parallel algoritmus
parallel computer – parallel (működésű) számítógép
parallel transmission – párhuzamos átvitel
parameter – paraméter
parameter list – paraméterlista
parameter transmission – paraméterátadás
parametric procedure – paraméterezett eljárás
parenthesis-free notation – zárójelmentes jelölés
parity – paritás
parity bit – paritásbit
parity check – paritás-ellenőrzés
parity error – paritáshiba
parse tree – leképzési fa
parsing – leképzés
partition – partíció
partitioned data base – particionált adatbázis
partitioned file – particionált [adat]állomány
partitioned memory management – particionált memóriakezelés
passive star – passzív csillag
password – jelszó
pattern matching – mintaillesztés
perforated card – lyukkártya
perforated tape – lyukszalag
peripheral device – periferikus mű
peripheral equipment – periféria
peripheral unit – periferikus egység
permanent memory – permanens memória
permanent virtual circuit – állandó virtuális összeköttetés
personal computer – személyi számítógép
phase – fázis
phrase structure grammar – mondatszerkezetű nyelvtan
physical address – fizikai cím
physical data base – fizikai adatbázis
physical data structure – fizikai adatszerkezet
physical file – fizikai [adat]állomány
physical layer protocol – fizikai rétegprotokoll
physical record – fizikai bejegyzés
physical relationship – fizikai kapcsolat
PIC (Position-Independent Code) – helyfüggetlen kód
PIC (Programmable Interrupt Controller) – programozható megszakításvezérlő
pictogram – piktogram
pipeline processing – futószalagszerű feldolgozás
pixel – képpont
plotter – rajzgép
pointer – mutató
pointer [data] type – mutató [adat]típus
point-to-point connection – ponttól pontig összeköttetés
postfix form – postfix forma
POL (Procedure-Oriented Language) – eljárásra orientált nyelv
polish form – lengyel forma
polling – lekérdezéses szervezés
polynomial check – polinom-ellenőrzés
polyphase merging – többfázisú összeválogatás
portability – hordozhatóság
position-independent code – helyfüggetlen kód
post-indexing – utóindexelés
post-mortem program – „post-mortem" program
post-paging – utólapozás
post-processing – utófeldolgozás
PPI (Programmable Peripheral Interface) – programozható perifériaillesztő
precedence grammar – elsőbbségi nyelvtan
precedence rule – precedenciaszabály
precedent function – előző függvény
precompiling – előfordítás
predefined [data] type – előre definiált [adat]típus
predicate – predikátum
prefix form – prefix forma
preindexing – előindexelés
prepaging – előlapozás
preprocessing – előfeldolgozás
presentation layer protocol – adatbemutató rétegprotokoll
preventive maintenance – megelőző karbantartás
primitive [data] type – primitív [adat]típus
printer – nyomtató
printing output area – nyomtatási mező
printing picture – nyomtatási sablon
priority – prioritás

priority allocation – prioritáskiosztás
privacy lock – hozzáférési zár
private [data] type – privát [adat]típus
privileged – kitüntetett
privileged instruction – privilegizált utasítás
privileged state – privilegizált állapot
problem-oriented language – problémaorientált nyelv
procedural language – procedurális nyelv
procedure – eljárás
procedure call – eljáráshívás
procedure call statement – eljáráshívó utasítás
procedure declaration – eljárásdeklaráció
procedure division – eljárási főrész
procedure module – eljárásmodul
procedure-oriented language – eljárásra orientált nyelv
process control – folyamatvezérlés
process control computer – folyamatvezérlő számítógép
processing unit – feldolgozóegység
processor – processzor
program – program
program block – programblokk
program certification – programhitelesítés
program-controlled machine – programvezérlésű gép
program counter – utasításszámláló regiszter
program editor – programszerkesztő
program execution – programfuttatás
program generator – programgenerátor
program interrupt – programmegszakítás
program library – programkönyvtár
programmable automaton – programozható automata
programmable interrupt controller – programozható megszakításvezérlő
programmable peripheral interface – programozható perifériaillesztő
programmable read-only memory – programozható permanens memória
programmable terminal – programozható terminál
programmed instruction – programozott oktatás
programmed resource – programozott erőforrás
programmer – programozó
programming – programozás
programming aids – programozási segédlet
programming environment – programozási környezet
programming error – programozási hiba
programming language – programozási nyelv
program optimization – programoptimalizálás
program package – programcsomag
program relocation – programáthelyezés
program run – programfuttatás
program section – programszakasz
program segment – programszegmens
program segmentation – programszegmentálás
program specification – programspecifikáció
program status word – programállapotszó
program testing – programtesztelés
program unit – programegység
program validation – programérvényesítés
program verification – programigazolás
program version – programváltozat
PROM (Programmable Read-Only Memory) – programozható permanens memória
prompt – fogadójel
proposition – ítélet
protection key – védőkulcs
protocol – protokoll
prunning – visszametszés
pseudo decimal digit – áltetrád
pseudo device – áleszköz
pseudocode – pszeudokód
pseudoinstruction – pszeudoutasítás
pseudolanguage – pszeudonyelv
pseudorandom numbers – pszeudovéletlen számok
PSTN (Public Switched Telephone Network) – nyilvános vonalkapcsolásos telefonhálózat
PSW (Program Status Word) – programállapotszó
public data network – nyilvános adathálózat
public switched telephone network – nyilvános vonalkapcsolásos telefonhálózat
punched card – lyukkártya
punched tape – lyukszalag
pure code – tiszta kód
push-down list – fordított lista
push-up list – direkt lista

PVC (Permanent Virtual Circuit) – állandó virtuális összeköttetés

Q

QAM (Queued Access Method) – sorbaállító adatelérési technika
QISAM (Queued Indexed-Sequential Access Method) – indexszekvenciális sorbaállító alap-adatelérési technika
QL (Query Language) – lekérdező nyelv
QSAM (Queued Sequential Access Method) – szekvenciális sorbaállító adatelérési technika
QTAM (Queued Telecommunication Access Method) – soros távfeldolgozási mód
quadratic sorting – kvadratikus rendezés
qualified expression – minősített kifejezés
qualified name – minősített név
qualifier – minősítő
quantization error – kvantálási hiba
quasi-parallel execution – kvázipárhuzamos végrehajtás
query file – szűrőállomány
query language – lekérdező nyelv, hivatkozási nyelv
queued access method – sorbaállító adatelérési technika
queued indexed-sequential access method – indexszekvenciális sorbaállító alap-adatelérési te
queued sequential access method – szekvenciális sorbaállító adatelérési technika
queued telecommunication access method – soros távfeldolgozási mód
quicksort – gyorsrendezés

R

radix – számrendszer alapja
radix notation – alapszámos írásmód
Radix-50 notation – Radix-50 ábrázolás
RAM (Random Access Memory) – véletlen hozzáférésű memória
random access media – közvetlen hozzáférésű adathordozó
random access memory – véletlen hozzáférésű memória
random access method – véletlen hozzáférési módszer
randomizing function – terítőfüggvény
random numbers – véletlen számok
random routing – véletlenszerű adatútképzés
raster display – raszteres kijelző
read head – olvasófej
reading – olvasás
reading error – beolvasási hiba
read-only memory – „csak olvasható" memória
real address – valódi cím
real constant – valós konstans
real [data] type – valós [adat]típus
real number – valós szám
real-time operation – valós idejű üzemmód, azonos idejű üzemmód
real-time system – valós idejű rendszer
record – rekord
record [data] type – rekord [adat]típus
recording density – tárolási sűrűség
recording rate – tárolási sebesség
recursion – rekurzivitás
recursive algorithm – rekurzív algoritmus
recursive data structure – rekurzív adatszerkezet
recursive procedure call – rekurzív eljáráshívás
reduced instruction set computer – korlátozott utasításkészletű számítógép
reductive syntactical analysis – összevonó szintaktikai elemzés
redundancy – redundancia
redundant code – redundáns kód
reentrancy – újraindíthatóság
reentrant program – újraindítható program
reference – hivatkozás
refreshing – frissítés
register – regiszter
regular grammar – reguláris nyelvtan
regular language – reguláris nyelv
relational data base – relációs adatbázis
relational expression – relációs kifejezés
relational operator – relációs operátor
relation condition – relációfeltétel
relative address – relatív cím
relative addressing – relatív címzés
relative error – relatív hiba
relay computer – jelfogós számítógép

reliability – megbízhatóság
relocatable binary form – áthelyezhető bináris forma
relocatable loader – áthelyezhető betöltőprogram
relocatable program – áthelyezhető program
relocation – áthelyezés
remote batch processing – távolsági kötegelt feldolgozás
reorganisation – újraszervezés
report generator/writer – táblázatgenerátor
reprogrammable read-only memory – újraprogramozható permanens memória
REPRON (Reprogrammable Read-only Memory) – újraprogramozható permanens memória
reserved word – lefoglalt szó
resolution – felbontás
resource – erőforrás
resource allocation – erőforrás-kiosztás
resource management – erőforrás-gazdálkodás
resource sharing – erőforrásosztás
response alphabet – válaszábécé
response time – válaszidő
restart – újraindítás
restart point – újraindítási pont
restructuring – újraszerkesztés
retention period – megőrzési periódus
return – visszatérés
return address – visszatérési cím
return code – visszatérési kód
reusable media – újrahasználható adathordozó
reverse video – fordított video
rewind – visszatekercselés
right justified representation – jobbra igazodó ábrázolás
RISC (Reduced Instruction Set Computer) – korlátozott utasításkészletű számítógép
RJE (Remote Job Entry) – távolsági kötegelt feldolgozás
roll back – visszagörgetés
roll-forward – előregörgetés
ROM (Read-Only Memory) – „csak olvasható" memória (permanens memória)
root segment – gyökérszegmens
round-off/rounding error – kerekítési hiba
routing – adatútképzés
routing strategy – bejárási stratégia
routing table – adatútképzési táblázat
running – futtatás
rule-based method – szabály alapú módszer
run-time error – végrehajtási hiba
run-time library – végrehajtási könyvtár

S

SAM (Serial Access Memory) – soros elérésű memória
SAM (Serial Access Method) – soros hozzáférési módszer
satellite computer – szatellita számítógép
scaling problem – léptékprobléma
scan code – letapogatási kód
scheduled maintenance – ütemezett karbantartás
scheduler – ütemező
scheduling algorithm – ütemezési algoritmus
scientific library – tudományos programkönyvtár
scrambler – bitsorkódoló
screen – képernyő
screen adapter – képernyőadapter
scrolling – görgetés
searching – keresés
search key – keresési kulcs
secondary channel – másodlagos csatorna
secondary memory – háttértár
second generation computer – második generációs számítógép
section – szakasz
sector – szektor
segment – szegmens
segmentation – szegmentálás
selection sorting – kiválasztásos rendezés
selective access – szelektív hozzáférés
selective file – szelektív [adat]állomány
selector channel – szelektorcsatorna
self-contained language – önálló nyelv
self-initializing program – önelőkészítő program
self-loading program – öntöltő program
semantic analysis – szemantikai analízis
semantic error – szemantikai hiba

semantic extension – szemantikai kiterjesztés
semantic invariance principle – jelentésmegmaradás elve
semantic rule – szemantikai szabály
semantics – szemantika
semaphore – szemafor
sentence – mondat
sentence symbol – fkezdőszimbólum
sentential form – mondatforma
separate compiling – külön fordítás
sequential access method – szekvenciális hozzáférési módszer
sequential algorithm – szekvenciális algoritmus
sequential file – szekvenciális [adat]állomány
sequential machine – szekvenciális gép
serial access media – soros hozzáférésű adathordozó
serial access memory – soros elérésű memória
serial access method – soros hozzáférési módszer
serial search – soros keresés
serial transmission – soros átvitel
service factor – kihasználási együttható
service program – segédprogram
session layer protocol – szessziós rétegprotokoll
set [data] type – halmaz [adat]típus
seven-layer reference model – hétréteges referenciamodell
S-expression – S-kifejezés
shared control – osztott vezérlés
shared resource – osztott erőforrás
Sheffer's function – Sheffer-függvény
Shell sort – Shell-rendezés
shifting – léptetés
side effect – mellékhatás
signal – jel
sign bit – előjelbit
sign condition – előjelfeltétel
significant digit – értékes számjegy
sign representation – előjel-ábrázolái
simple condition – egyszerű feltétel
simple name – egyszerű név
simplex channel – szimplex csatorna
simplex communication – szimplex összeköttetés
simplex method – szimplex módszer
simulation – szimuláció
simulation language – szimulációs nyelv
simulation-mode update – szimulációs aktualizálás
simulation program – szimulációs program
simultaneous operation – szimultán működés
slave processor – alárendelt processzor
slow memory – lassú memória
software – szoftver
software engineering – programozási technológia
software house – szoftverház
software resource – szoftver erőforrás
SOH (Start OF Heading) – fejléckezdet karakter
sorting – rendezés
sorting program – rendezőprogram
source language – forrásnyelv
source library – forrásnyelvű könyvtár
source program – forrásprogram
special-purpose computer – célszámítógép
special-purpose operating system – speciális célú operációs rendszer
spell checking – helyesírás-ellenőrzés
spooling – háttérnyomtatás
spreadsheet – táblázatkezelő
stack – verem
stack operation – veremművelet
stack-oriented computer – veremorientált számítógép
stack pointer – veremmutató
standard function – standard függvény
standard label – szabványos címke
standard reference system – standard hivatkozási rendszer
standard subroutine – standard szubrutin
standard writing – szabványírás
stand-by redundancy – tartalék redundancia
start-of-heading character – fejléckezdet karakter
start-of-text character – szövegkezdés karakter
star transmission network – csillaghálózat
state diagram – állapotdiagram
synchronous computer – szinkron (működésű) számítógép
static attribute – statikus attribútum
static data structure – statikus adatstruktúra
static expression – statikus kifejezés

static model – statikus modell
static subroutine – statikus szubrutin
statement – utasítás
status information – állapotinformáció
status register – állapotregiszter
status word – állapotszó
step-by-step execution – lépésenkénti végrehajtás
stepwise refinement principle – lépésenkénti finomítás elve
stochastic model – sztochasztikus modell
stored program – tárolt program
strategic game – stratégiai játék
string – jelsorozat
srting [data] type – füzér [adat] típus
strongly-typed language – erősen típusos nyelv
structured [data] type – strukturált [adat]típus
structured program – strukturált program
structured programming – strukturált programozás
STX (Start of TeXt) – szövegkezdés karakter
subprogram – alprogram
subrange [data] type – intervallum [adat]típus
subroutine – szubrutin
subroutine library – szubrutinkönyvtár
subscript – index[1]
subscripted variable – indexes változó
subscript overflow – indextúlcsordulás
successor function – következő függvény
supervisor – felügyelőprogram, szupervizor
supervisor call instruction – felügyelőprogram-hívó utasítás
supervisor state – felügyelő állapot
suspended program – felfüggesztett program
SVC (Switched Virtual Circuit) – kapcsolt virtuális összeköttetés
SW (Status Word) – állapotszó
switched line – kapcsolt vonal
switched network – kapcsolt hálózat
switched virtual circuit – kapcsolt virtuális összeköttetés
switching-node computer – csomóponti kapcsoló számítógép
symbol – szimbólum
symbol dictionary – szimbólumtáblázat
symbolic addressing – szimbolikus címzés
symbolic instruction – szimbolikus utasítás
symbolic reference – szimbolikus hivatkozás
symbol manipulation language – szimbólumfeldolgozó nyelv
symmetric list – szimmetrikus lista
symmetric matrix – szimmetrikus mátrix
SYN (SYNchronization character) – szinkronizálási karakter
synchronization – szinkronizálás
synchronization character – szinkronizálási karakter
synchronization information – szinkronizálási információ
synchronous – szinkron
synchronous channel – szinkron csatorna
synchronous communication – szinkron adatátvitel
static variable – statikus változó
synchronous network – szinkronhálózat
syntactical ambiguity – szintaktikai kétértelműség
syntactical analyser – szintaktikai elemző
syntactical analysis – szintaktikai analízis
syntactic error – szintaktikai hiba
syntactic extension – szintaktikai kiterjesztés
syntax – szintaxis
syntax category – szintaktikai kategória
syntax diagram – szintaxisdiagram
syntax-directed compiler – szintaxisvezérelt fordítóprogram
syntax tree – szintaxisfa
synthetized attribute – szintetizált attribútum
system analysis – rendszerelemzés
system crash – rendszerösszeomlás
system design – rendszerfejlesztés
system device – rendszereszköz
system disk – rendszerlemez
system file – rendszerállomány
system generation – rendszergenerálás
system library – rendszerkönyvtár
system log – rendszernapló
system macroinstruction – rendszer-makroutasítás
system program – rendszerprogram
system programmer – rendszer-programozó
system software – rendszerszoftver
system variable – rendszerváltozó

T

table – táblázat
table look-up instruction – táblázatban kereső utasítás
table search – keresés táblázatban
tailored information system – igényre szabott információs rendszer
tape operating system – mágnesszalagos operációs rendszer
tape puncher/printer – szalaglyukasztó
target language – tárgynyelv
target program – tárgyprogram
task – taszk, feladat
task control block – feladatvezérlő blokk
task management – feladatkezelés
TCAM (TeleCommunication Access Method) – távfeldolgozási mód
TCB (Task Control Block) – feladatvezérlő blokk
TDM (Time-Division Multiplexing) – időosztásos multiplexelés
teaching machine – oktatógép
telecommunication access method – távfeldolgozási mód
teleprocessing – táv-adatfeldolgozás
teleprocessing peripheral equipment – távvezérelt periféria
teletex – teletex
telex – telex
terminal – terminál
terminal equipment – adatvégállomás, végberendezés
terminal station – adatállomás
terminal symbol – terminális szimbólum
testing – tesztelés
testing data base – tesztelési adatbázis
test generator – tesztgenerátor
test program – tesztprogram
text editor – szövegszerkesztő
text mode – szöveges mód
third generation computer – harmadik generációs számítógép
thrashing – lapcséplés
threaded code – láncolt kód
throughput – áteresztőképesség
time-division multiplexing – időosztásos multiplexelés
time-division system – időosztásos rendszer
time slice – időszelet
token ring – vezérjelgyűrű
tolerable error – megengedett hiba
top-down method – felülről lefelé haladó módszer
top-down syntactical analysis – felülről lefelé haladó szintaktikai elemzés
TOS (Tape Operating System) – mágnesszalagos operációs rendszer
tournament sorting – kieséses rendezés
toy problem – játékfeladat
trace – nyomkövetés
track – sáv
track index – sávindex
traffic matrix – forgalmi mátrix
transaction – tranzakció
transaction-oriented processing – tranzakciófeldolgozás
transfer instruction – átviteli utasítás
translatable expression – áthelyezhető kifejezés
transparent data – transzparens adat
transparent resource – transzparens erőforrás
transport layer protocol – átviteli rétegprotokoll
trap – csapda
tree – fa
truncating – csonkítás
truth table – igazságtáblázat
truth value – igazságérték
TSS (Time Sharing System) – időosztásos rendszer
tuning – hangolás
turtle graphics – teknőcgrafika
two-address computer – kétcímű számítógép
two-way list – kétirányú lista
type compatibility – típusegyezés

U

unconditional control instruction – feltétlen vezérlésátadó utasítás
undefined file – szerkezet nélküli [adat]állomány
underflow – alulcsordulás
universal quantifier – univerzális kvantor
unpacked decimal representation – zónázott decimális ábrázolás
unpacking – kicsomagolás

updating – aktualizálás
user – felhasználó
user guide – felhasználói kézikönyv
user indetification code – felhasználói azonosító kód
user program – felhasználói program
utility program – kezelőprogram

V

validity region – érvényességi tartomány
value – érték
variable – változó
variable length record – változó hosszúságú rekord
variable-partitions method – változó partíciók módszere
VDL (Vienna Definition Language) – bécsi definíciós nyelv
vector display – vonalrajzoló kijelző
verb – ige
version number – verziószám
video disk – videolemez
Vienna definition language – bécsi definíciós nyelv
virtual address – virtuális cím
virtual computer – virtuális számítógép
virtual data item – virtuális adatelem
virtual machine – virtuális gép
virtual memory – virtuális memória
virtual memory addressing – virtuális memóriacímzés
virtual telecommunications access method – virtuális távközlési hozzáférési módszer
virtual terminal – virtuális terminál
virtual terminal station – virtuális adatállomás
virus – vírus
visibility scope – láthatósági tartomány
VOL (VOLume label) – adathordozói címke
volatile memory – felejtőmemória
volume – kötet, táregység
volume label – adathordozói címke
von Neumann machine – Neumann-gép
V-series recommandations – V-szabványok
VTAM (Virtual Telecommunication Access Method) – virtuális távközlési hozzáférési módszer

W

waiting – várakozás
waiting model – várakozási modell
waiting queue – várakozási sor
waiting strategy – várakozási stratégia
waiting system – várakozási rendszer
waiting time – várakozási idő
WAN (Wide Area Network) – kiterjedt hálózat
warm start – melegindítás
Warnier's method – Warnier-módszer
wide area network – kiterjedt hálózat
wildcard character – generatív karakter
window – ablak
word – szó
word length – szóhosszúság
word processor – szövegszerkesztő
word size – szóhosszúság
word structure – szószerkezet
working set – munkahalmaz
working storage – munkarekesz
workstation – munkaállomás
writing error – kiírási hiba

X

XOR (eXclusive OR) – kizáró VAGY
XPS (eXPert System) – szakértőrendszer
X-series recommendations – X-szabványok

256,- FT

Az Alaplap Könyvek sorozat kötetei

1. Farmosi István—Kis János—Szegedi Imre: Víruslélektan
2. Nagy Gábor: Tömör gyönyör
3. Kis János—Szegedi Imre: Új víruslélektan
4. Kis János—Szegedi Imre: Vírushatározó
5. Számítástechnikai alaplexikon I. (Jodál Endre: Általános fogalmak)

A SZÁMÍTÁSTECHNIKAI ALAPLEXIKON tervezett köteteinek fő témakörei

I. Általános fogalmak
II. Adatkommunikáció
III. Gazdaság, pénzügy, kereskedelem
IV. Ipari számítástechnika
V. Eszközök és gyártási technológiák
VI. Mesterséges intelligencia
VII. Négynyelvű szótár